FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Engagement of Traditionally 2D Web Applications Presented Using 3D Games

João Pedro da Costa Ribeiro

Master in Informatics and Computing Engineering

Supervisor: Prof. António Fernando Vasconcelos Cunha Castro Coelho

March 31, 2023

Engagement of Traditionally 2D Web Applications Presented Using 3D Games

João Pedro da Costa Ribeiro

Master in Informatics and Computing Engineering

March 31, 2023

Abstract

The advancement of technology has enabled the implementation of 3D graphics on web technologies, allowing for the creation of immersive 3D web applications. The use of gamification in interactive experiences, such as teaching and work simulations, has been widely studied and has shown an increase to user engagement and motivation.

In the last few years, with this advancement in technology, many still recur to the typical websites when presenting information. But the people behind such websites want more user engagement. This can be seen with the rise of 3D renders of products on commercial websites and immersive 3D experiences for artist websites where, by incorporating 3D technology, they retained users for longer and, in turn, got free publicity by word of mouth.

This dissertation aims to apply both 3D technology and gamification principles on typically 2D websites. To that end, it reviews the current state of the art in 3D web technology and gamification, and their effects on user experience, which, individually, both show promising results in user engagement. It then aims to investigate the impact of presenting traditionally 2D content in a 3D gamified environment by prototyping a curriculum website and collecting metrics and feedback through a questionnaire at the beginning and end of the user experience, with a focus on the perspective of recruiters. To obtain this information, the website needs to be dynamic and allow for the presentation of any topic of information. Thus, several tests with variations based on metrics and feedback were made, to verify that the conclusions are general and accurate.

The main contributions of this thesis are the study of the effects of 3D technology on user engagement, methods of presenting 2D information in such an environment, a boilerplate and setup for other developers in the open-source community and a few tools to be released as libraries to ease the development of similar experiences (such as 3D text wrapping, a SVG modeler and JSON model generation).

The thesis aims to draw conclusions on how to better present 2D content in a 3D gamified environment, and provide insights on the impact of this approach on user engagement, motivation, and overall user experience. The findings of the paper will contribute to the current understanding of gamification and 3D web technology, and provide practical recommendations for designers and developers looking to implement similar approaches in their projects.

Keywords: web game, 3D Website, gamification, user engagement

Resumo

O avanço da tecnologia permitiu a implementação de gráficos 3D na maioria das tecnologias web, permitindo a criação de aplicações web 3D imersivas. O uso da gamificação em experiências interativas, como o ensino e simulações de trabalho, tem sido amplamente estudado e demonstrou um aumento na interação e motivação do utilizador.

Nos últimos anos, apesar desse avanço na tecnologia, muitos ainda recorrem a websites típicos ao apresentar informações. Mas os responsáveis por esses sites pretendem que o utilizador se envolva mais. Isso pode ser visto com o aumento de renderizações 3D de produtos em websites comerciais e experiências 3D imersivas em sites de artistas onde, ao incorporar a tecnologia 3D, eles retêm os utilizadores por mais tempo e, por sua vez, obtêm publicidade boca-a-boca gratuita.

Esta dissertação tem como objetivo aplicar tanto a tecnologia 3D como os princípios de gamificação em websites tipicamente 2D. Para isso, existe uma revisão do estado atual da arte em tecnologia web 3D e gamificação, e seus efeitos na experiência do utilizador que, individualmente, ambos mostram bons resultados no envolvimento do utilizador. Também visa investigar o impacto da apresentação de conteúdo tradicionalmente 2D num ambiente gamificado em 3D, ao prototipar um site de currículo e a recolher métricas e feedback por meio de questionários no início e no final da experiência do utilizador. Para obter esta informação, o website necessita de ser dinâmico e possibilitar a apresentação de qualquer tema de informação. Assim, vários testes com alterações entre eles com base nas métricas e feedback foram feitos, para verificar que as conclusões são gerais e acertadas.

As principais contribuições desta tese são o estudo dos efeitos da tecnologia 3D no envolvimento do utilizador, métodos de apresentação de informações 2D em tal ambiente, uma base e pré-configuração para outros desenvolvedores da comunidade open-source e algumas ferramentas a serem lançadas como bibliotecas para facilitar o desenvolvimento de experiências semelhantes (como formatação de texto 3D, um modelador SVG e geração de modelos dinâmica a partir de ficheiros JSON).

A tese tem como objetivo tirar conclusões sobre como apresentar melhor o conteúdo 2D num ambiente gamificado em 3D, e fornecer insights sobre o impacto dessa abordagem no engajamento, motivação e experiência geral do utilizador. As descobertas da dissertação contribuirão para a compreensão atual da gamificação e da tecnologia web 3D, e fornecerão recomendações práticas para designers e desenvolvedores que procuram implementar abordagens semelhantes nos seus projetos.

Palavras-chave: jogo web, website 3D, gamificação, envolvimento de utilizadores

Acknowledgements

I'd like to thank **prof. António Coelho** for his contributions towards the evaluation steps of this project.

I'd also want to thank, as my mother and friend, **Manuela Ribeiro** for supporting me in every manner as a single mother since i was a baby. Your love and support made me who I am today.

Thank you to my close family, for always helping me every time it was needed. I was only able to achieve my goals with your help.

My thanks go to **Machoado** for the many afternoons of venting, sharing and support, that helped me grow as a person.

Appreciate all of the **The Boys**, that gave me unlimited experiences, days of fun and of support for over a decade now.

Above all, thank you **Inês Oliveira** for, not only the help with the 3D models, but also the support and motivation given during this last stretch of my academic journey, and for the continued emotional support given in very hard times for me. It goes without saying but I am incredibly glad to have you by my side. Love you <3

João Ribeiro

“Sem pão, sem sal“

João Ribeiro

Contents

1 Introduction **1**
1.1 Context . . . 1
1.2 Motivation . . . 2
1.3 Objectives . . . 2
1.4 Contributions . . . 3
1.5 Document Structure . . . 3

2 State of the Art **4**
2.1 Web Technology . . . 5
2.1.1 Adobe Flash . . . 6
2.1.2 Unity . . . 7
2.1.3 JavaScript Frameworks . . . 8
2.1.4 Web APIs . . . 9
2.2 Gamification . . . 11
2.2.1 Gamification in 3D . . . 12
2.2.2 Frameworks . . . 12
2.2.3 Principles . . . 13
2.3 User Experience . . . 14
2.3.1 Impact . . . 15
2.3.2 Principles . . . 15
2.4 3D Technologies . . . 16
2.4.1 Three.js . . . 16
2.4.2 Babylon.js . . . 17
2.4.3 PlayCanvas . . . 17
2.4.4 A-Frame . . . 18
2.5 Existing 3D Web Applications . . . 18
2.5.1 3D Renders in a 2D Website . . . 18
2.5.2 Fully 3D Website . . . 19
2.5.3 3D Game Website . . . 19
2.6 Metaverse . . . 19
2.7 Summary . . . 20

3 Methodology **21**
3.1 Problem Statement . . . 21
3.2 Proposed Solution . . . 22
3.2.1 Topic . . . 22
3.2.2 Game Design . . . 23
3.2.3 Technologies . . . 24

3.3 Methodology . . . 24

4 The Ice Above - An Interactive CV Experience **26**

4.1 Background . . . 26
4.2 Game Concept and Theme . . . 26
4.3 Target Audience . . . 27
4.4 Gameplay . . . 27
4.4.1 Game Loop . . . 32
4.4.2 Player Controls . . . 33
4.4.3 Screens . . . 33
4.5 Resources . . . 35
4.5.1 3D Assets . . . 35
4.5.2 Animations . . . 35
4.5.3 Audio . . . 35
4.5.4 Thee.js . . . 35
4.6 Summary . . . 36

5 Implementation and Tools Developed **37**

5.1 File Structure . . . 37
5.2 Game Architecture . . . 38
5.2.1 Approach . . . 38
5.2.2 Canvas . . . 38
5.2.3 Classes . . . 39
5.3 Interface . . . 51
5.4 Forms . . . 52
5.5 Metrics . . . 53
5.6 Tools Developed . . . 53

6 Evaluation **54**

6.1 Early Iterative Steps . . . 54
6.2 Pre-Game Form . . . 55
6.2.1 Questions . . . 55
6.2.2 Results . . . 56
6.3 Post-Game Form . . . 59
6.3.1 Questions . . . 59
6.3.2 Results . . . 60
6.4 Time Spent Metric Results . . . 65
6.5 Analysis and Comparison . . . 66
6.5.1 User Satisfaction . . . 66
6.5.2 User Engagement . . . 67
6.5.3 User Retention . . . 68
6.6 Conclusion . . . 68

7 Conclusions and Future Work **69**

7.1 Project Conclusions . . . 69
7.2 Research Conclusions . . . 70
7.3 Future Work . . . 71

References **72**

A Pre-Game Questionnaire 76
A.0.1 Questions 76
A.0.2 Results 79

B Post-Game Questionnaire 83
B.0.1 Questions 83
B.0.2 Results 87

List of Figures

2.1 Adobe Flash Google Search Trend [2] 7
2.2 Gamification Google Search Trend [3] 11
2.3 UI / UX Google Search Trend [10] 15
2.4 Three.js Google Search Trend [5] 17

3.1 Design Science Diagram [?] 24

4.1 Main Screen 27
4.2 Start Area 28
4.3 Credits and Instructions 28
4.4 Start Area - Narrow Tunnel 29
4.5 Main Area - Tunnels and Signs 29
4.6 Tunnel 30
4.7 Player Swinging Pick-Axe 31
4.8 Prompt After the Ice Wall Breaking 31
4.9 Book Detailed Information 32
4.10 Character Dancing 32
4.11 Loading Screen 34

5.1 Code Structure 38
5.2 Class Structure 39
5.3 Character Class Relational Diagram 48
5.4 Main Screen with Start Prompt 51
5.5 Initial Book Prompt and End Form Prompt 52
5.6 Wall Break Book Prompt 52

6.1 Start Form Question 1 56
6.2 Start Form Question 2 56
6.3 Start Form Question 3 57
6.4 Start Form Question 4 57
6.5 Start Form Question 5 58
6.6 Start Form Question 6 58
6.7 Start Form Question 7 59
6.8 Start Form Question 8 59
6.9 End Form Result 1 61
6.10 End Form Result 2 61
6.11 End Form Result 3 62
6.12 End Form Result 4 62
6.13 End Form Result 5 63

6.14 End Form Result 6 63
6.15 End Form Result 7 64
6.16 End Form Result 8 64
6.17 End Form Result 9 65
6.18 Time Spent Graph 66
6.19 Average Time Spent on Websites per Industry [14] 67

A.1 Start Form Question 1 76
A.2 Start Form Question 2 76
A.3 Start Form Question 3 77
A.4 Start Form Question 4 77
A.5 Start Form Question 5 77
A.6 Start Form Question 6 78
A.7 Start Form Question 7 78
A.8 Start Form Question 8 78
A.9 Start Form Question 1 79
A.10 Start Form Question 2 79
A.11 Start Form Question 3 80
A.12 Start Form Question 4 80
A.13 Start Form Question 5 81
A.14 Start Form Question 6 81
A.15 Start Form Question 7 82
A.16 Start Form Question 8 82

B.1 End Form Question 1 83
B.2 End Form Question 2 84
B.3 End Form Question 3 84
B.4 End Form Question 4 85
B.5 End Form Question 5 85
B.6 End Form Question 6 85
B.7 End Form Question 7 86
B.8 End Form Question 8 86
B.9 End Form Question 9 86
B.10 End Form Result 1 87
B.11 End Form Result 2 87
B.12 End Form Result 3 88
B.13 End Form Result 4 88
B.14 End Form Result 5 88
B.15 End Form Result 6 89
B.16 End Form Result 7 89
B.17 End Form Result 8 89
B.18 End Form Result 9 90

List of Tables

Abbreviations and Symbols

CV	Curriculum Vitae
3D	Three Dimensional
FOV	Field of View
POV	Point of View
FBX	a type of 3D model file created using the Autodesk FBX software
JSON	JavaScript Object Notation
JS	JavaScript
Tech	Technology
SVG	Scalable Vector Graphics
TTF	TrueType Font
NPM	Node Package Manager
Lerp	Linear Interpolation

Chapter 1

Introduction

1.1 Context

The internet has become an essential part of modern life, and it is used for a wide range of purposes, including communication, entertainment, and education. With the proliferation of web applications and websites, users have come to expect dynamic and interactive content that engages them in meaningful ways. As such, website designers and developers are constantly searching for new and innovative ways to make their sites more appealing to users. [17]

The prevalence of two dimensional websites for the past three decades have left them trivial to the average user. As such, with the value of user retention, innovative ways to attract people to ones website became important. While there are many ways to tackle this subject that already have proven successful, most shy away from the full use of interactive 3D environments. [12]

Also, over the past few years, there has been a rapid evolution of hardware technology, resulting in faster processors, more efficient graphic cards, and increased storage capabilities. This has enabled developers to create more demanding web applications, such as 3D games, that were once only possible on native applications. The emergence of powerful tools like Three.js and Babylon.js, which utilize WebGL and other modern web technologies, has enabled developers to create high-quality web games and applications that provide immersive and interactive experiences.[4, 13]

However, this evolution of hardware technology has also presented new challenges, particularly in the area of user experience. As web applications become more complex and resource-intensive, it becomes increasingly difficult to provide seamless and enjoyable user experiences. Furthermore, the shift in user expectations resulting from the hardware evolution has resulted in users demanding more sophisticated and visually impressive web applications. This presents developers with new challenges, as they must ensure that their applications meet these higher expectations while maintaining a balance between performance and user experience.

Traditionally, web applications and websites have been presented in a 2D format, with users interacting with text, images, and other static content. However, the new web technologies and better hardware have made it possible to easily present web applications and websites in a 3D format. This offers an entirely new level of interactivity and engagement for users and opens up a whole new realm of possibilities for website designers and developers, which I explore.

1.2 Motivation

The prevalence of two-dimensional websites for the past three decades has left them trivial to the average user, resulting in many potentially amazing content being ignored or overlooked. This is especially true for individuals who lack the visual design skills to create engaging websites, causing great ideas to be lost in the vast sea of the internet.

Through this thesis, I hope to provide the groundwork for more innovative and compelling web applications with the use of these new technologies.

In addition, the potential of 3D gamified content on the web is immense and presents a unique opportunity to create engaging user experiences. As an area that combines two of my favorite areas in programming, I strive to prove its usefulness and make it more widespread. Fun experiences can be beneficial to all, regardless of the objective behind it.

Ultimately, the aim of my study is to investigate whether 3D web games can increase user engagement, enjoyment, and time spent on websites that traditionally present 2D content. By doing so, I hope to contribute to the body of knowledge on web design and development, providing insights that can be used to improve the user experience of web applications and websites.

1.3 Objectives

The main objective of this study is to investigate the potential benefits of using 3D gamified content on web applications and websites that traditionally present content in a 2D format. This study aims to contribute to the body of knowledge on web and game development and web architecture by providing insights into the effectiveness of using 3D gamified content to increase user experience, engagement, and time spent on websites.

To achieve this objective, the study will start by reviewing the literature on the use of 3D gamified content in web design and development, as well as the theoretical foundations of user engagement and experience. This will provide a theoretical framework for the study, as well as a comprehensive understanding of the existing research on this topic.

Then, a template 3D gamified game will be developed using the knowledge acquired from the research. This website will be adaptable to different contexts and show the potential of the chosen approach. It will also have to generate content from 2D information, as its goal is to enable the easy conversion existing 2D websites into 3D websites (by taking developer/user created 2D information in the JSON format).

The study will then use a mixed-methods approach to collect and analyze data on user experience, engagement, and time spent this website compared to those that use traditional 2D content. The study will use both quantitative and qualitative data collection methods, including surveys, user testing, and metrics from users.

The results of the study will be used to provide insights into the effectiveness of using 3D gamified content in web design and development, as well as recommendations for improving the user experience of web applications and websites.

1.4 Contributions

This project intends to research ways to increase user experience and engagement by using a 3D gamified approach. To that end, a template website was developed and iterated upon. Hence, the main contributions are:

- Analysis on what 3D gamified aspects improve user experience and engagement.
- Development of a dynamic 3D web game that demonstrates the aforementioned capabilities.
- Development of tools to facilitate the creation of these applications.
- Creation of guidelines for 3D gamified websites.

1.5 Document Structure

After the contextualization of the problem and the objectives of the research, this dissertation starts by exploring the State of the Art of previous web innovations, gamification, user experience and it's effects on engagement, 3D web technology, current 3D web applications on Chapter 2.

Then, on Chapter 3, the methodology is explained, where I describe the problem at hand and the proposed solution taking into account the previous research.

Afterwards, a full elaboration of the game's theme, loop, gameplay, controls and resources is presented on Chapter 4.

On Chapter 5 I delve into the games architecture, coding structure, technology and deployment.

Penultimately, Chapter 6 shows the forms results and the metrics collected and it's respective evaluation.

Finally, in Chapter 7 the conclusions from the thesis are drawn, encompassing the potential, limitations and possible direction of future developments.

Chapter 2

State of the Art

This section of the thesis is very extensive as the topics it covers are diverse. It is sectioned into 7 parts:

- **Web Technology** As part of understanding where the 3D technology can build upon and differentiate from past attempts, an exploration of previous technology that tried to innovate in the areas of user engagement and experience is a must.
- **Gamification** Since this topic is already quite researched, it should be reviewed as many of the conclusions from previous papers could be applied in this new context.
- **User Experience** As one of the main end goals is to improve, in a general sense, the user experience, an extensive analysis on current standards is warranted.
- **3D Web Technology** The project needs to be built upon existing technology and frameworks that support 3D, so an investigation on pros and cons of the available options is needed.
- **Existing 3D Web Applications** This thesis has the objective of researching the potential of applications of 3D to present 2D content. As such, it is necessary to dive into what current web applications do and what they don't.
- **Related Work** This section aims to show how this thesis contributes to the broad computer science body of research.

2.1 Web Technology

In the last 2 decades, the web has evolved at a staggering rate. Since 2003, the number of websites increased by 45 times, reaching 1.6 billion in 2018 [32, 13].

Web technology advanced together with this demand, to the point that all frameworks and engines used today are completely different from what existed just 10 years go. [24]

"Since the 2000s' the number of different gadgets with the specific screen dimensions people use has been multiplied. Developers had a need for adapting the application to all devices & screens accordingly. The further the technical progress went, the more monotonous job they had." [13]

The term "Web 1.0" is used to describe the first generation of the World Wide Web, which was characterized by static HTML pages that were viewed in a web browser. Web 1.0 was primarily a one-way communication medium, with content creators publishing information and users consuming it. The focus was on creating a web presence for businesses and organizations, rather than on interactive or social features. The rise of Web 2.0 marked a significant shift in web development, with the introduction of dynamic, user-generated content, social media, and web applications. Web 2.0 sites such as Facebook, Twitter, and YouTube enabled users to interact with each other and share information in real-time.

Web 3.0, also known as the "Semantic Web," is the next phase in the evolution of the World Wide Web. It is characterized by a focus on intelligent search, data sharing, and machine-to-machine communication. The Semantic Web is built on the idea of adding meaning to data on the web, so that computers can understand and interpret it. This enables advanced search capabilities and new applications, such as personal assistants, that can understand and respond to natural language queries.

3D web technologies are a relatively new addition to the web, and they can be seen as a part of the Web 3.0 movement. They allow for interactive and immersive experiences on the web, bringing a new level of depth and realism to web applications. While 3D technologies have been around for a while, they were previously limited to specialized software and hardware. With the rise of WebGL and other web standards, 3D graphics can now be rendered directly in a web browser, making it accessible to a wider audience. [18, 37]

In terms of their place in the evolution of the web, 3D technologies can be seen as a natural progression from the interactive and social features of Web 2.0. They offer new opportunities for user engagement and interaction, and can be used to create innovative new applications and experiences. As web technologies continue to evolve, it is likely that we will see even more advanced and immersive 3D experiences on the web, further blurring the lines between virtual and physical reality.

In this section I will explore the most meaningful technologies regarding the topic of web innovation and content presentation across the this century.

2.1.1 Adobe Flash

"It feels gross and disgusting to say it out loud, but the death of Flash might actually be a bad thing for creativity. Developers and designers loved Flash because it enabled audacious ideas that didn't necessarily require huge amounts of coding knowledge." [12]

Adobe Flash was a multimedia software platform that was widely used for creating animations, games, and interactive content for the web. The software was developed by Macromedia in the mid-1990s and was acquired by Adobe Systems in 2005. It was a popular tool for web designers and developers for over two decades.

Adobe Flash was initially developed as a tool for creating interactive animations and small games for the web. The platform used a scripting language called ActionScript to enable interactivity in these animations and games. Over time, Adobe Flash evolved into a more powerful platform, capable of delivering rich media content, including audio and video.

Adobe Flash was used extensively in the early days of the internet for delivering multimedia content. It was particularly popular for creating web-based games and interactive experiences. Many popular games and applications, such as Farmville, Club Penguin, and Homestar Runner, were built using Adobe Flash.

One of the key strengths of Adobe Flash was its cross-platform compatibility. The platform was designed to work across different operating systems and web browsers. This made it possible to deliver rich multimedia content to a wide range of devices and platforms.

Despite its many strengths, Adobe Flash had several limitations. One of the most significant was its performance. As multimedia content became more complex and demanding, Adobe Flash struggled to keep up. This led to performance issues and stability problems, particularly on mobile devices.

Another significant limitation of Adobe Flash was its security vulnerabilities. The platform was frequently targeted by hackers and cybercriminals, who exploited weaknesses in the software to deliver malware and other types of malicious content.

In 2017, Adobe announced that it would be discontinuing Flash by the end of 2020. The decision was driven by several factors, including the rise of mobile devices, which had made Adobe Flash less relevant, and the growing concerns around security and performance issues.

As a result of the discontinuation, many websites and applications that relied on Adobe Flash had to be redesigned or replaced. Adobe has encouraged developers to migrate their content to newer, more modern platforms, such as HTML5, which offer better performance, security, and cross-platform compatibility.

In conclusion, Adobe Flash was a multimedia software platform that played a significant role in the early days of the internet. It was widely used for creating interactive content and games for the web. However, the platform had several limitations, including performance and security issues. As a result, Adobe made the decision to discontinue Flash in 2020, and developers have been encouraged to migrate their content to newer, more modern platforms.

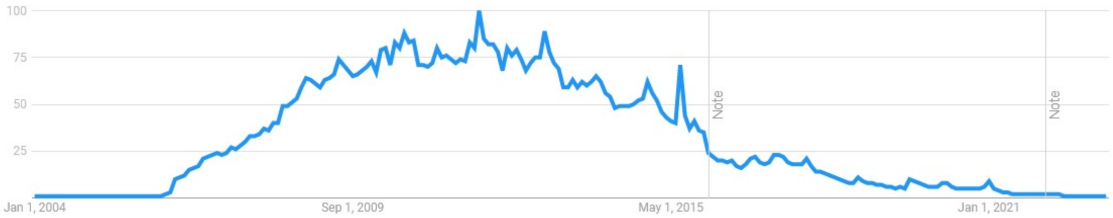


Figure 2.1: Adobe Flash Google Search Trend [2]

2.1.2 Unity

Unity is a powerful game engine that has been widely used in the creation of video games and interactive experiences. However, Unity has also been utilized in web development as a platform to create rich, interactive, and immersive 3D web experiences. The Unity engine is a cross-platform game engine that allows developers to create games and interactive experiences for various platforms including desktops, mobile devices, and even virtual reality (VR) headsets.

Unity for web development provides several capabilities that allow for the creation of highly interactive and engaging 3D web experiences. One of the primary capabilities of Unity is its ability to render high-quality 3D graphics in real-time. This allows for the creation of highly detailed and immersive 3D environments that can be explored by users in real-time. Additionally, Unity provides a robust physics engine that allows for the creation of realistic physics-based interactions within these 3D environments.

Another key capability of Unity 3D for web development is its support for scripting in C#. This allows developers to create highly interactive and dynamic experiences that can be customized and modified based on user input and other events.

Unity also provides a comprehensive set of tools for creating and managing complex game logic, including support for animations, artificial intelligence (AI), and physics simulations. These tools can be leveraged to create highly engaging and challenging gameplay experiences that can be customized to the specific needs and goals of the web experience.

One of the primary limitations of using Unity for web development is its relatively large file sizes. 3D web experiences created with Unity can often be quite large, which can lead to slow loading times and a poor user experience. Additionally, Unity requires a relatively high level of technical expertise and programming knowledge, which can make it challenging for novice developers to get started with the platform.

Despite these limitations, Unity remains a powerful and versatile platform for creating rich and immersive 3D web experiences. With its advanced rendering capabilities, robust physics engine, and support for scripting in multiple programming languages, Unity provides developers with the tools they need to create engaging and interactive web experiences that can captivate users and drive engagement.[24]

2.1.3 JavaScript Frameworks

2.1.3.1 React

React is a widely popular JavaScript library for building user interfaces that was developed by Facebook. It was first introduced in 2013 and has since then become one of the most popular choices for building dynamic and responsive web applications.

At its core, React provides a declarative way of building UI components that are reusable, maintainable, and easy to reason about. React is based on the concept of component composition, where UI components can be nested and composed in a tree-like structure to create complex user interfaces. This makes it easy to manage the complexity of large-scale applications and to build reusable UI components that can be shared across multiple projects.

One of the key advantages of React is its virtual DOM (Document Object Model), which provides a lightweight representation of the actual DOM. This allows React to efficiently update the UI by selectively re-rendering only the components that have changed, rather than re-rendering the entire page. This makes React highly performant and suitable for building large-scale, data-intensive applications.

In recent years, React has continued to evolve with new features and improvements. React Hooks, which were introduced in 2018, provide a new way of writing reusable and composable stateful logic in functional components. React also has strong support for server-side rendering and has made significant strides in improving accessibility and internationalization support.

React has a large and active community of developers and is widely used by companies of all sizes, including Facebook, Netflix, Airbnb, and Instagram. Its popularity is reflected in the vast number of third-party libraries, tools, and plugins that are available for React, making it a versatile and flexible tool for building web applications. Overall, React's ease of use, scalability, and performance make it a go-to choice for building complex user interfaces.

2.1.3.2 Angular

Angular is an open-source front-end web application framework that was initially released in 2010 by Google. The framework is designed to develop dynamic, single-page web applications and mobile applications. Angular uses TypeScript, a superset of JavaScript, to create clean and readable code. Its main features include dependency injection, two-way data binding, and declarative templates.

One of the most significant advantages of Angular is its modularity, allowing developers to break down complex applications into smaller, manageable pieces. This makes Angular ideal for large-scale enterprise applications.

In recent years, Angular has undergone a significant transformation, transitioning from AngularJS to Angular, which is a complete rewrite of the original framework. Angular is now a more efficient and modern framework, providing better performance, improved scalability, and better support for mobile development.

Some of the most notable developments in recent years include the introduction of the Angular CLI, which simplifies the process of creating, testing, and deploying Angular applications, and the introduction of Angular Universal, which enables server-side rendering of Angular applications.

The community around Angular is vast and active, with many contributors constantly working on the development of new features, extensions, and libraries. This community support ensures that Angular remains relevant and up-to-date with the latest technologies and industry trends.

According to the State of JS survey in 2020, Angular remains a popular choice for web developers, ranking among the top three most used front-end frameworks. Its popularity is reflected in the many companies and organizations that use Angular, including Google, Microsoft, IBM, and Forbes.

2.1.3.3 Vue

Vue.js is a progressive JavaScript framework that is used for building user interfaces. It was developed by Evan You in 2014 and has since become a popular tool for front-end development. Vue is similar to other popular JavaScript frameworks like React and Angular, but is known for its simplicity and ease of use.

Vue is based on the Model-View-ViewModel (MVVM) architecture pattern, which allows developers to separate the user interface from the application logic. This makes it easier to manage and maintain code, especially in larger projects. Vue is also designed to be reactive, which means that the interface automatically updates in response to changes in the underlying data.

One of the key features of Vue is its flexibility. Developers can choose to use Vue as a standalone library, or integrate it with other libraries and frameworks. Vue also has a large ecosystem of plugins and extensions, which makes it easy to add functionality to an application.

Vue has seen significant growth and development in recent years. In 2020, the Vue team released Vue 3, which included several major updates and improvements. One of the most significant changes was the introduction of a new rendering system called the Composition API, which provides a more flexible and modular way of building components.

In terms of community size and popularity, Vue has a large and active community of developers. It is estimated that over 3 million websites are built with Vue, and there are over 160,000 Vue-related repositories on GitHub. Vue is also used by many popular companies and organizations, including Alibaba, Xiaomi, and GitLab.

Overall, Vue is a powerful and versatile tool for building user interfaces. Its simplicity and flexibility make it a popular choice for developers of all skill levels, and its active community ensures that it will continue to be a relevant and useful technology for years to come.

2.1.4 Web APIs

API technology, or Application Programming Interface, allows web developers to integrate various functionalities and features into their websites or applications. APIs provide a standardized way for different software components to communicate with each other, enabling developers to easily incorporate external resources into their projects.

One of the primary uses of APIs is to add media and social features to websites. For example, APIs allow developers to easily integrate video and audio players, image galleries, and other media components into their sites. APIs can also be used to integrate social media features like sharing, commenting, and following. These social features have been shown to drastically improve user engagement on websites and applications.

One popular example of API integration is the embedding of YouTube videos on websites. By using the YouTube API, developers can easily embed videos into their sites, customize the player interface, and even control video playback programmatically. Another popular use of APIs is the integration of Facebook comments and login functionality. This allows users to easily log in and comment on content using their Facebook accounts, increasing user engagement and social sharing.

In the past, blog public chatrooms were a popular social feature on websites. These chatrooms allowed users to engage in real-time conversations with each other, creating a sense of community and increasing user engagement. While blog chatrooms have become less popular in recent years, they remain a relic of the past and serve as an example of how social features can enhance user engagement on websites.

2.1.4.1 Youtube

The YouTube API has had a significant impact on user engagement and experience by allowing websites and applications to embed YouTube videos seamlessly.

With the YouTube API, developers can customize the player to match the look and feel of their website or application, control playback options, and track usage data. This level of customization allows for a more seamless integration of videos and enhances the user experience.

In addition, the YouTube API also allows for the integration of social features, such as commenting and sharing, which further increases user engagement. By allowing users to share and comment on videos directly from a website, the YouTube API creates a sense of community and encourages user participation.

Research suggests that websites that incorporate videos have an average user engagement time of around 6 minutes, which is around 6x the average for websites.

2.1.4.2 Facebook

The Facebook API allows for the integration of Facebook features into websites, such as the ability to log in with a Facebook account and add Facebook comments to articles. This integration can significantly enhance the user experience by providing a familiar and convenient way for users to engage with the content on the site.

One of the main advantages of Facebook API integration is that it eliminates the need for users to create a new account and remember a separate username and password for each site. This can increase user retention and engagement as users are more likely to return to a site that offers a seamless login experience. Additionally, Facebook comments can increase user engagement by allowing users to express their opinions and interact with others on the site.

However, there are also potential drawbacks to Facebook API integration. For example, some users may be hesitant to log in with their Facebook accounts due to privacy concerns. Additionally, relying too heavily on Facebook for user engagement can limit a site's independence and control over its user experience.

2.1.4.3 Blog Chats

Blog chats, also known as public chatrooms, were a popular feature in the early days of the internet when blogs were a prominent form of online communication. They were essentially online chatrooms that were embedded within a blog or website, where visitors could engage in real-time conversations about specific topics or interests.

Blog chats were seen as a way to foster community engagement and interaction, as they provided a space for readers to connect and share their thoughts with one another. They were often moderated by the blog owner or other designated individuals to ensure that conversations remained civil and on-topic.

However, with the rise of social media platforms and instant messaging apps, the popularity of blog chats has declined significantly. Many websites and blogs have removed this feature altogether or replaced it with other forms of community engagement, such as forums or social media groups.

2.2 Gamification

Gamification is the application of game design elements and mechanics to non-game contexts in order to engage users and encourage participation. It can be used to increase motivation, build loyalty, and enhance the overall user experience. The use of gamification has become increasingly popular in recent years, especially in the areas of education, healthcare, and business. [40, 46, 50, 57, 15]

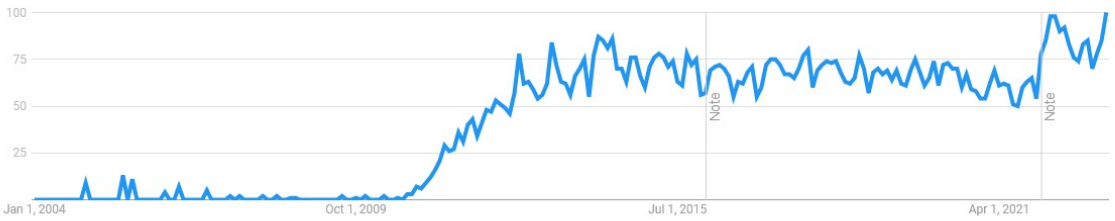


Figure 2.2: Gamification Google Search Trend [3]

2.2.1 Gamification in 3D

Gamification is the process of integrating game mechanics and design elements into non-game environments such as websites, apps, and marketing campaigns to increase user engagement, motivation, and loyalty. It dates back to the early 1900s when game-like elements were used in educational environments to make learning more fun and interactive [40, 33, 34, 21, 49, 48]. But gamification as we know it today emerged in the early 2000s, when technological advancements and social media led to an explosion of online gaming and virtual worlds. Researchers and practitioners are beginning to explore the potential of gamification to improve user experience and inspire desired behaviors [27, 28, 51]. Today, gamification is a rapidly growing field with many applications in various industries such as education, healthcare, retail and finance [39, 35]. It involves using various game mechanics and design elements such as points, badges, leaderboards, challenges, rewards and social interactions to create an engaging and immersive experience that meets users' intrinsic motivations and psychological needs. Research shows that gamification increases user engagement, motivation, and satisfaction, as well as improves learning outcomes, productivity, and revenue. [40, 30, 53]

A major difference between gamification in 3D and 2D is the level of immersion and realism. Gamification in 3D provides a more immersive and interactive experience, allowing users to explore virtual environments, interact with 3D objects, and engage in realistic simulations (users can practice and learn in a simulated environment before applying their knowledge in real life). This sense of immersion can lead to higher levels of engagement and motivation as users feel more connected to the experience and can visualize achieving their goals. In contrast, gamification in 2D can still be effective in driving engagement and motivation, but may not be as immersive or realistic. [23, 22, 36, 41]

This makes gamification a powerful tool for web developers and businesses looking to increase user engagement and motivation. With the advent of 3D web technologies, the potential for immersive and interactive gaming experiences is greater than ever, giving businesses new opportunities to connect with audiences and drive desired behaviors. As the field continues to evolve, there will be new innovative ways in which gamification can be used to create engaging and meaningful experiences for users. [32]

2.2.2 Frameworks

Gamification frameworks provide a structured approach to designing and implementing gamification in a product or service. These frameworks are developed based on research and experience and provide guidance on various aspects of gamification, such as user motivation, feedback mechanisms, and game mechanics. [52, 16, 36]

One example of a gamification framework is the Octalysis Framework developed by Yu-kai Chou. It is based on the concept of eight core drives that motivate people to take action, such as curiosity, social influence, and empowerment. The framework provides a visual tool for designers to analyze and design gamified experiences based on these core drives.

Another example is the MDA (Mechanics, Dynamics, Aesthetics) framework developed by Robin Hunicke, Marc LeBlanc, and Robert Zubek. It breaks down games into three components: the mechanics or rules of the game, the dynamics or how the mechanics interact with each other, and the aesthetics or emotional responses elicited by the game. This framework provides a way to analyze and design gamification experiences based on these components.

Frameworks like these provide a starting point for designers to create effective gamification experiences. However, it is important to note that each framework may not be suitable for every project or situation. Designers should evaluate the specific needs and goals of their project and choose a framework that aligns with those goals. Additionally, frameworks should be adapted and customized as needed to fit the unique needs of the project and the users.

2.2.3 Principles

Theoretically, by applying the core principles of gamification to the website's design, it results in a overall improvement on the users engagement with the website.

As such, these are the main principles of gamification that can be applied in this context:

2.2.3.1 Reward Systems

Providing users with rewards or incentives for completing tasks or reaching milestones is a key part of gamification. This can include badges, points, or other virtual rewards. [50]

2.2.3.2 Feedback

Users need to receive feedback and recognition for their efforts. This can include positive reinforcement, such as praise or virtual rewards, as well as constructive feedback to help them improve.

This also includes any type of visual and audio feedback, as it has been proven that small audio cues and visual effects when interacting drastically improves the users perception. [50]

2.2.3.3 Clear Goals

In order for gamification to be effective, users need to have clear goals and objectives to work towards. This helps to keep them motivated and engaged.

2.2.3.4 Progression

Users need to be able to feel some progress and see how far they have come. This helps to motivate them and provides a sense of achievement.

2.2.3.5 Social Interaction

Adding a social element to gamification can enhance user engagement. This can include leaderboards, social sharing, and collaboration with other users. [29, 30]

2.2.3.6 Personalization

Personalization refers to the process of tailoring experiences to the individual user. It can include elements such as avatar customization, personalized recommendations, and adaptive difficulty levels.

2.3 User Experience

User experience refers to the overall experience a person has while interacting with a product, service, or system. In web design, UX is essential for creating positive interactions between users and the website. Good UX design considers the user's goals, abilities, and expectations to create a website that is easy to use, efficient, and enjoyable. [42, 32, 55, 17, 19, 25]

Evaluating UX is an important step in the design process as it ensures that the final product is intuitive, usable and enjoyable for the end-user. One of the most common methods for evaluating UX is through usability testing. Usability testing involves observing users as they perform tasks on the product and gathering feedback on the usability and effectiveness of the design. This can be done through user interviews, surveys, or even remote testing sessions. [45, 44]

Another method for evaluating UX is through analytics. Analyzing user behavior data, such as click-through rates, bounce rates, and time on site, can provide valuable insights into how users are interacting with the product. This data can help identify areas of the product that need improvement, and guide the design team towards making data-driven decisions to improve the overall user experience.

Heuristic evaluation is also a popular method for evaluating UX. This involves evaluating the design against a set of established usability principles or "heuristics" that have been shown to improve user experience. Examples of these principles include visibility of system status, consistency and standards, and error prevention. Heuristic evaluation can be done by a UX expert, a group of experts, or even the design team themselves. [55, 45]

Finally, user feedback and surveys can also be used to evaluate UX. Collecting feedback directly from users through surveys, feedback forms, or focus groups can provide valuable insights into how users are experiencing the product. This feedback can be used to identify common pain points or areas of confusion, and guide the design team towards making improvements to the product. [45, 44]

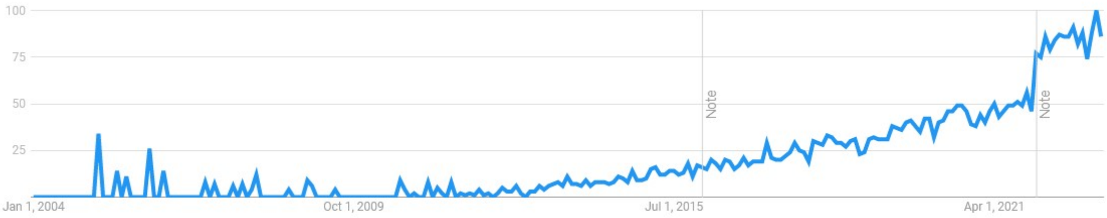


Figure 2.3: UI / UX Google Search Trend [10]

2.3.1 Impact

When a website has a good user experience, it can positively impact the users' perception of the brand or person, and increase the likelihood of return visits. It can increase engagement, encourage them to spend more time on the website, improve the chances of converting visitors into customers, and increase the chances of return. Conversely, poor UX design can lead to frustration, confusion, and a lack of trust in the website, which can drive users away. [47, 26, 20]

As the objective of this thesis, from a research point of view, is to prove there are benefits for using 3D technology to present 2D content, the past research results indicate that a good UX is imperative for its success.

2.3.2 Principles

Theoretically, by applying the core principles of user experience to the game, interface and visual design, it results in a overall improvement on the users perception of the website.

As such, these are the main principles of UX that can be applied in this context:

2.3.2.1 Accessibility

Accessibility is an essential UX principle that aims to ensure that a website is usable by people with disabilities. This can include providing alternative text for images, providing transcripts for videos, and designing the website with a clear and easy-to-use interface.

2.3.2.2 Consistency

Consistency refers to designing the website with a consistent look and feel throughout the user's journey. This includes using the same fonts, colors, and design elements across all parts of the website. Consistency helps users to navigate the website more easily and reduces cognitive load.

2.3.2.3 Visual hierarchy

Visual Hierarchy refers to the arrangement of visual elements on the website to guide the user's attention and provide a clear understanding of the content hierarchy. This can include the use of font size, color, and placement to make important information stand out.

2.3.2.4 Usability

Usability refers to the ease of use of the website. A website with good usability is easy to navigate and understand, and the user can quickly find what they are looking for. Usability can be achieved through the use of clear labels, intuitive navigation, and a simple layout. [44, 43]

2.3.2.5 Memorability

Memorability refers to the user's ability to remember the website and return to it. A website that is memorable has a unique design and features that stick in the user's mind. This can be achieved through the use of branding, unique features, and a memorable user experience.

2.3.2.6 Aesthetics

Aesthetics refer to the visual appeal of the website. Aesthetically pleasing websites can create a positive emotional response in the user and increase engagement. This can be achieved through the use of attractive design elements, visually appealing color schemes, and engaging content.

2.4 3D Technologies

2.4.1 Three.js

Three.js is a JavaScript library that allows developers to create and display 3D graphics in web browsers.It was first released in 2010 and has since become one of the most widely used 3D libraries on the web. Three.js provides a simple and efficient way to create complex 3D scenes without the need for extensive knowledge of computer graphics or 3D programming, using WebGL. [**?**]

In the last five years, three.js has continued to evolve and expand its capabilities. Some major developments include support for virtual reality (VR) and augmented reality (AR) through WebVR and WebXR APIs. This allows developers to create immersive 3D experiences through VR headsets and other AR devices. [54, 24]

Another important development is the physical-based rendering (PBR), which allows for more realistic lighting and shading in 3D scenes. This opened up new opportunities to create high-quality, photo-realistic 3D graphics in the browser.

One of three.js' strengths is its open source community, which has contributed numerous extensions and extensions to the library. These tools include physics simulation, terrain generation, and machine learning tools.The community has also created many examples and tutorials to help developers learn new techniques and start using three.js. [18, 31]

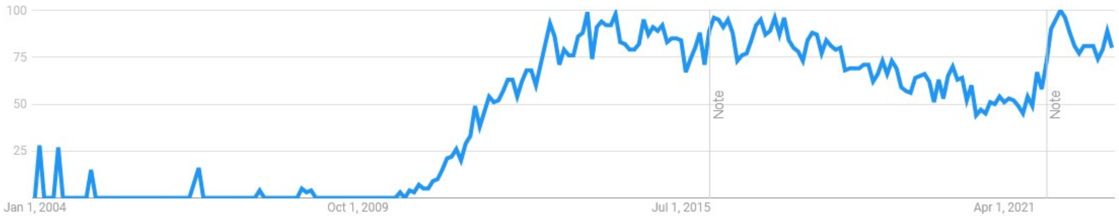


Figure 2.4: Three.js Google Search Trend [5]

2.4.2 Babylon.js

Babylon.js is a powerful 3D game engine and rendering framework written in JavaScript, built on the WebGL graphics API. It allows you to create complex 3D scenes and interactive games in the browser. In recent years, Babylon.js has made many important advances,including improved performance and stability, as well as extended physics simulation, animation and special effects features.One of Babylon.js' unique features is its online text editor and compiler, which allows developers to setup and test this technology quickly and easily without installing software locally. Babylon.js also has a small but dedicated open source community, which has contributed many plugins and add-ons to further extend its functionality. Although there is a smaller community than three.js, Babylon.js is still a popular choice for the development of web-based 3D games and applications. [**?**]

2.4.3 PlayCanvas

PlayCanvas is a browser-based game development platform that allows users to create 3D games and apps using a combination of HTML5, WebGL, and JavaScript. It offers a powerful and intuitive interface that allows users to create complex game logic without writing code.

One of the main features of PlayCanvas is its real-time collaboration system, which allows multiple users to work on the same project simultaneously. This feature is particularly useful for game development teams, as it allows members to work together seamlessly from anywhere in the world.

Despite its impressive features and capabilities, PlayCanvas is a relatively small and niche technology, with a smaller community compared to other game development platforms such as Unity or Unreal Engine. However, it has gained a dedicated following among indie game developers and hobbyists due to its ease of use and accessibility. PlayCanvas is also open-source, which means that users can contribute to its development and create custom tools and extensions for the platform.

2.4.4 A-Frame

A-Frame is an open source web framework that allows developers to create virtual reality experiences on the web. It is a relatively new technology that was first introduced in 2015 and is currently maintained by the Mozilla VR team. A-Frame's main feature is ease of use, as it allows developers to create VR experiences using an HTML-like syntax, making it accessible to web developers who may not have more advanced 3D graphics programming experience.

One of the main advantages of A-Frame is that it is based on Three.js, a popular 3D graphics library, which allows it to take advantage of its capabilities while providing a more streamlined development process. A-Frame also provides several pre-built components, including support for controllers and other input devices, as well as physics simulation and collision detection.

A-Frame isn't as widespread in terms of community size and popularity as some other web VR technologies like Three.js or Babylon.js, but it still has a dedicated community of developers actively contributing to the project. Participate and create a new experience with it. The framework has been used to create a variety of VR experiences, including games, educational simulations, and architectural visualizations.

One of the unique features of A-Frame is its focus on accessibility and inclusivity, with features such as support for screen readers and a focus on creating experiences for users with disabilities. The framework also emphasizes open standards and interoperability, making it easy to integrate with other networking technologies and tools.

Overall, A-Frame is a promising technology for creating VR experiences on the web, especially for those who are already familiar with web development and want to try VR without learning a whole new programming language or toolset. Its ease of use and focus on accessibility and compatibility make it an attractive option for developers looking to create engaging and inclusive virtual reality experiences.[18]

2.5 Existing 3D Web Applications

Here I will go over a few examples of existing similar websites that use 3D technology in some way. This provides some groundwork on whats the norm and what users expect when interacting with 3D components.

2.5.1 3D Renders in a 2D Website

Chirpley.ai is a good example of such. Through the use of a typical responsive and smooth scrollable page layout, it keeps it's familiarity to users who are not used to 3D environments, while providing a reactive 3D background to keep the user on its feet.

It presents all the information in a typical 2D manner, while keeping the 3D aspects to the background and extra context to the information.

While this works beautifully, it is already quite common in the industry.

2.5.2 Fully 3D Website

Admireamaze.debijenkorf.nl tries to tell a story in 3D using a very similar technique that old 2D platformer games used, where the user follows the character (a bee in this case) while it moves in a bi-dimensional plane.

This makes the website extremely accessible as the progression triggered like a single page 2D website: through the use of scrolling. While this is a major upside, on the other hand the website feels very slow and cumbersome to read the information, as it provides very little engagement and interaction.

2.5.3 3D Game Website

Playkeepout.com is a retro Doom style game that puts the player in a progressive dungeon crawler. It is heavily focused on performance, having no toll on even older hardware.

The assets are very basic, akin to 90's games. It also incorporates clear audio and visual feedback in each interaction of the game.

This makes the game, while simple in concept, very enjoyable to play and, with its accessibility due to being available on a website, very easy to go back and replay during someones schedule gaps.

2.6 Metaverse

The metaverse is a term coined to describe a shared, virtual space that is created by the convergence of virtual reality, augmented reality, and the internet. It is a 3D world that can be accessed and interacted with by anyone, from anywhere in the world. The concept of the metaverse has been around for several decades, but recent advancements in technology have made it a more realistic possibility. The development of 5G networks, virtual reality hardware, and 3D web technologies have made it possible to create immersive, real-time, shared virtual environments that can be used for work, play, and social interaction.

In recent years, there has been a surge of interest in the metaverse, particularly in the gaming industry. Companies like Roblox, Fortnite, and Minecraft have built massive, immersive virtual worlds that have attracted millions of users. These virtual worlds are not just for gaming, however. They are also being used for virtual events, concerts, and even business meetings. The metaverse is seen as the next step in the evolution of the internet, where users can interact with each other and with digital content in a more immersive and engaging way.

It is becoming increasingly clear that 3D web technologies will play a critical role in its development. The metaverse is envisioned as a fully immersive and interactive virtual space that is designed to replicate the real world in terms of social interaction, economy, and culture. To achieve this level of immersion and interactivity, the metaverse needs advanced 3D graphics and rendering capabilities, which can be provided by 3D web technologies.

The metaverse has the potential to change the way we work, play, and interact with each other online. It offers new opportunities for businesses to connect with customers and for individuals to connect with each other. However, there are also concerns about the impact of the metaverse on privacy, security, and the digital divide. As the metaverse evolves, it will be important to ensure that it is accessible to all, and that users have control over their own data and experiences. The development of the metaverse is still in its early stages, but it is an exciting and rapidly evolving area of web development.

2.7 Summary

To summarize, 3D technology shows lots of promise as a technology.

In the last years, many technologies emerged and gave developers plenty of tools to be creative and create better experiences for the user. However, many of them were deprecated for their flaws in favor of tried and tested methods and their respective tech. This is what I attribute as the main cause of the simplistic and normalized current 2D web experience, where only small deviations from the norm take place.

Advancements in the area of the user experience set some key principles that allow such simple and standardized to still be pleasant to use. These principles shaped our web ecosystem.

However, the rise in popularity of gamification proved a demand for a new method of presenting content. Gamification shows that, if well implemented, it can provide an increase in most user metrics, from engagement and retention to satisfaction. While presenting some new key principles to reach these results, it still goes in line with the user experience research, but applied in a different context.

As such, with the advancement of hardware, most devices have been able to run very demanding web apps for a while. This meant that the need for incredibly simple websites for the sake of performance was not a must anymore, as devices now can handle much more load. This enabled 3D browser-based technology to come to fruition and provide told for developers to explore that area. Three.js is currently the moste popular.

This all supports the thesis topic, that 2D content can now be presented in fully 3D environments with gamification elements, and that such would be an overall positive experience for the user and increase most user metrics.

Chapter 3

Methodology

3.1 Problem Statement

As analysed in the previous chapter, most technology has converged into responsive paged 2D design. The previous contenders for the outlet of creativity have either been discontinued or became too outdated to use. As such, 3D technology is now the biggest contender for major innovation regarding the user experience.

The issue lies on the fact that no good web game engine exists to facilitate the development of these apps. Most of them only offer tools as renderers with little available tools to expand as games.

Not only that, but the nonexistent research on the use of 3D environments to present 2D content show a huge unexplored area with a lot of potential.

So, with no research to back the use of 3D technology to present 2D content and the lack of easy to use tools to program 3D web games, a problem arises.

To tackle this problem, the solution must address the following:

- Present 3D content in an engaging and enjoyable manner, as per the User Experience research suggested. This means it must be accessible, consistent, memorable, learnable and have a clear visual hierarchy. This can be very difficult as these paradigms usually only apply to app and web development, which are mainly 2D. Nonetheless that does not mean they are inapplicable in a new environment, it just requires a more attentive approach than usual.

- Engage the user with gamification elements, as per the gamification research suggested. This means introducing progression and feedback systems, while maintaining clear goals for the experience. While gamification has been thoroughly researched, the topic of using an actual 3D game approach with these principles in mind is groundbreaking, so new ways of applying them will be necessary.

- Allow for the easy switching or alteration of the content presented, just like a typical 2D website does. To this end, the solution must be generated in some way from 2D information. This does not mean the solution has to be totally procedurally generated, in fact it shouldn't, as we want as much control as possible from the experience to guarantee consistency and quality. As such, only some required parts should be generated from text, as the 3D models needed to set the theme are better made outside or taken from free repositories.
- Be easily replaced regarding the 3D models used, to be personalized according to the makers preferences and theme. This means the specific game loop and architecture should be independent of the models used. The whole game structure should be modular and scalable, to be able to use each piece as needed without having to adapt each module. While this is not noticeable on the final project, this makes a huge difference on the viability of whats covered on the next item.
- Streamline the process to create gamified 3D websites, as currently these tools are very scarce and create difficulties for the adoption of such technology. Hence, architecture and specific missing tools should be developed. These tools should work as a framework add-on and provide groundwork for important functions that do not yet exist.
- Analyse, through forms and metrics, the impact this type of website has compared to typical 2D websites. This means that there must be detailed forms to evaluate preconceptions and postconceptions, and the tracking of important information regarding the use of the website and further comparison to reach theoretical conclusions to contribute to the general knowledge.

3.2 Proposed Solution

To tackle all the previous items regarding the problem statement, the proposed solution consists on the creation of a dynamically generated 3D game experience website, that implements gamification principles while being accessible to people who are not used to 3D environments. During the development, wherever possible, create and separate specific tools that are viable taking into account the scope of the project.

3.2.1 Topic

As for the topic that the website covers, this could be anything that is typically depicted in a 2D manner:

- **Sales:** Sales-oriented 2D websites are a common way for businesses to showcase and sell their products or services. However, these websites often lack innovation and creativity, relying on generic templates and layouts that fail to engage users. Moreover, the integration of extensive product information, images, and reviews requires a complex and extensive database, making the development and maintenance of such sites challenging and costly.

As such, while it would be a good candidate, it would only work as a template for the objective of this thesis, so a static type website would be preferable.

- **Educational:** Gamification and education have evolved hand-in-hand for the past decade and, with the need for interactivity and engagement, this would be almost perfect to use as the topic for the thesis. The problem arises when thinking about the questionnaire, as it could influence the questionnaires more than whats desirable based on how much the specific topic would interest the specific tester, regardless of background or preconceptions. As such, a more neutral topic would be better.

- **Exposure:** Exposure websites are some of the most stagnant by sheer volume and template-driven design. However, it is also where most the 3D efforts have been concentrated after education and simulation. While this could be a suitable topic to choose, due to the existing attempts of modernization I chose the next topic.

- **Portfolio / Curriculum:** Finally, Portfolios and Curriculum websites stand out as the best candidate. Having the fame of being single-page cookie cutter apps, with the single purpose of exposing projects and experience, they remain incredibly important. They are almost always 2D, simplistic, light in information, need to be memorable and, the website itself, can count towards the value of the curriculum as a project. Not only that but, since it's a CV website, there is less likely to exist a strong bias towards the content.

Hence, the topic of a CV website was chosen for the purpose of the research.

3.2.2 Game Design

As for the theme of the website and game, any can work, it depends on personal taste, so it will not be taken into account in this dissertation.

Regarding game design, as the focus is on engagement and enjoyment, the most important aspects would be immersion, engagement, interaction and effects. With this in mind, several choices were made:

- **Point of view**: The 3 big contenders are First Person, Third Person and Asymmetrical(Top-Down View). In this order, immersion is stronger in First Person POVs and weaker on Asymmetrical POVs. By that logic, First Person would be the best but, when taking into account ease of navigation and spatial awareness, its the opposite. This is quite important as the target audience, recruiters, do not have to be well versed in gaming controls and 3D navigation. A compromise between immersion and ease of navigation seems best, thus opting for a Third Person POV.

- **Level of Interaction** Level of Interaction, meaning the amount of action types needed to achieve a goal, add, as the name suggests, more interaction to an app. But, as interaction increases, the ease of access to, in this case, information decreases, as they pose as a mechanical and time barrier. As this type of website is simple in nature and should be easy to access, more so than other websites, at max 2-3 levels of interaction are ideal. Any more than that can become cumbersome and deter users.

3.2.3 Technologies

The technologies chosen are a mix of existing popular frameworks, being Vite, React and Three.js. Vite was chosen for its setup and development facilitation. React was chosen as the frontend framework as it has direct integration with Three.js, even though they were avoided in favor of developing broader JS tools.

While Babylon.js was the first chosen during the research and early to mid point of development, I had to scrap it all in favor of Three.js. Babylons strengths lie in its ease of setup and development as they provide a web app to code and render, together with many examples. However, that's where it's strengths end, as it is newer than the veteran Three.js and is not Open-Source, meaning that the community and groundwork existing for this technology does not compare. Due to this, during development I found myself having to reinvent the wheel many times for things Three.js already could do from the get go. Three.js may not be as streamline and clean, but being open-source was proven more valuable.

Finally, as this is a static website, a back end framework was not necessary.

3.3 Methodology

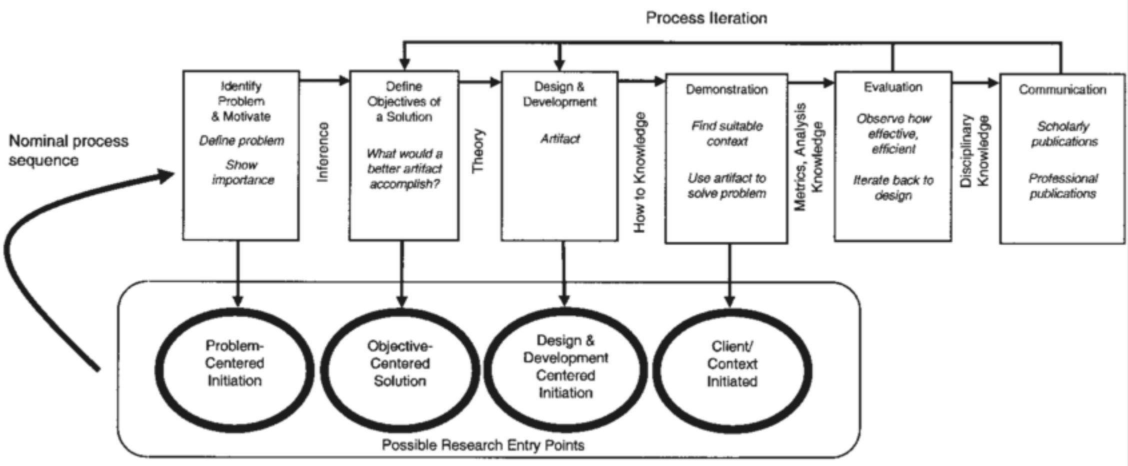

Process Iteration
Nominal process sequence
Identify Problem & Motivate
Define problem
Show importance
Inference
Define Objectives of a Solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How to Knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, Analysis Knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary Knowledge
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 3.1: Design Science Diagram [**?**]

Design science is an approach to problem-solving that focuses on creating solutions using a combination of scientific principles and creative design methods. It is an iterative process that involves identifying a problem, researching possible solutions, creating a prototype, testing and evaluating the solution, and refining the design based on user feedback. This approach is often used in engineering, information systems, and other fields where a systematic approach to problem-solving is required.

One of the primary benefits of design science is that it provides a structured framework for solving complex problems. By breaking down the problem into smaller parts and systematically testing potential solutions, design science allows for a more efficient and effective approach to problem-solving. Additionally, by involving users in the design process, it ensures that the solutions are tailored to meet their needs and preferences.

However, there are also some limitations to design science. One potential drawback is that it can be a time-consuming and resource-intensive process, particularly when involving multiple iterations of design and testing. Additionally, design science can sometimes lead to solutions that are overly complex or difficult to implement in practice.

Despite these limitations, design science can be highly applicable to a single person project that depends on user feedback. By following a systematic approach to design and testing, a single person can create a prototype that meets the needs of their users while also ensuring that the final product is efficient and effective. Furthermore, by incorporating user feedback into the design process, the final product is more likely to be successful in the market, leading to increased engagement and user satisfaction.

Chapter 4

The Ice Above - An Interactive CV Experience

4.1 Background

The Ice Above is a dynamically generated 3D web gamified CV created to showcase and study the potential and effects of presenting typically 2D content in a 3D gamified manner. Everything from the game design, architecture, loop, theme and tools were built from scratch, using only vanilla three.js (with react and vite but none of the framework specific functionalities were used.

The models were also either custom modeled or picked from free repositories and the animations were imported using Mixamo, as, while incredibly important towards the project, it was not the focus.

4.2 Game Concept and Theme

As an arctic explorer that fell down a hole (see Figure 4.1, where the POV is from said hole), the player now has to try and escape this icy cave. To that end, the player has a pick-axe and a journal, both crucial to his information gathering. As the player wanders through the cave exploring, he will find rows of transparent icy walls that contain several information from a previous explorer that resided here. By breaking these walls, the player then learns and writes the findings on his journal, hoping that some of this information proves useful to his escape.

The style chosen was sort of low-poly, as, to be able to create a dynamically generated game that could run on the web, the model's size could not be too large.

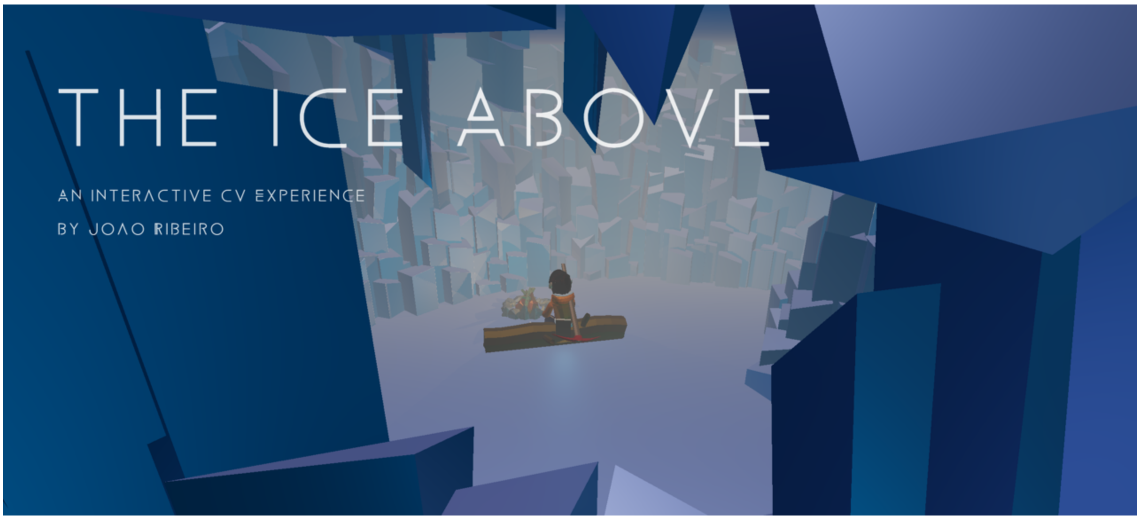


Figure 4.1: Main Screen

4.3 Target Audience

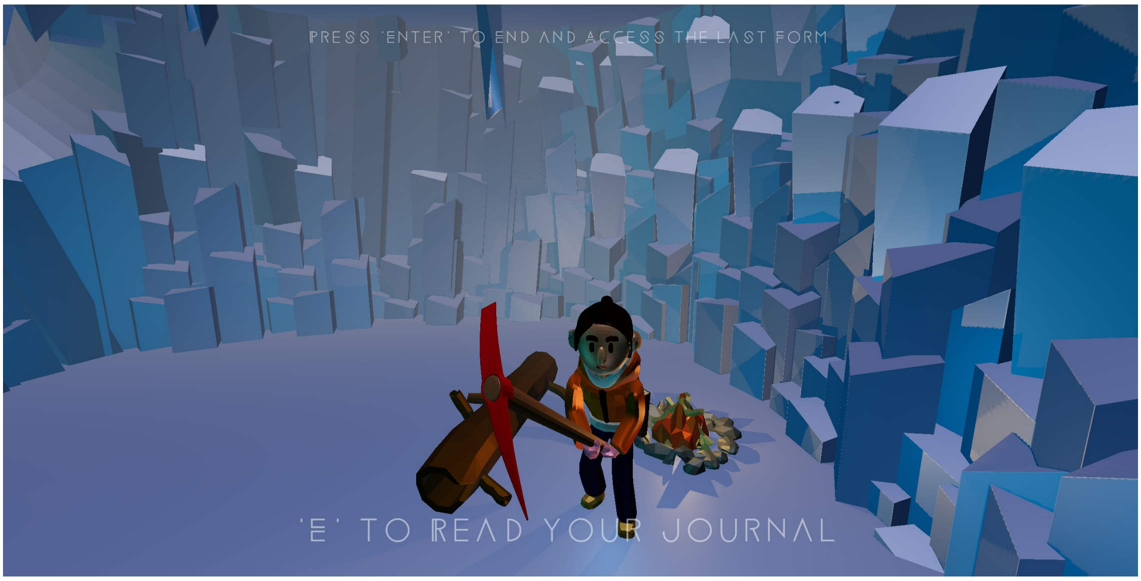
The target audience for this game are recruiters, more specifically tech recruiters as the CV chosen was my own. However, the actual target audience encompasses anyone curious to know more about both the technology and me.

This is both good and bad. Since the target audience is relatively narrow and niche, it should be easier to pinpoint key details and needs that this type of user might need. However, other than the necessity for ease of access to information, everything else depends entirely on the users previous game experience and spacial awareness. This makes it incredibly hard to decide which factors matter more regarding the target audience so, as research aims to provide information and groundwork for the future, less emphasis was put on performance.

4.4 Gameplay

To begin the game, the user must lock the mouse on the screen (deliberate choice to help with computer immersion) and press *Space*. This cues a set of animations on the character to stand up and equip his gear and the camera goes from stationary to the typical smooth third person camera.

The game is composed of a single level, divided in two sections. The first section is extremely small and serves as a small buffer for the user to get used to the controls, akin to a tutorial. It is accessible right after starting the experience.

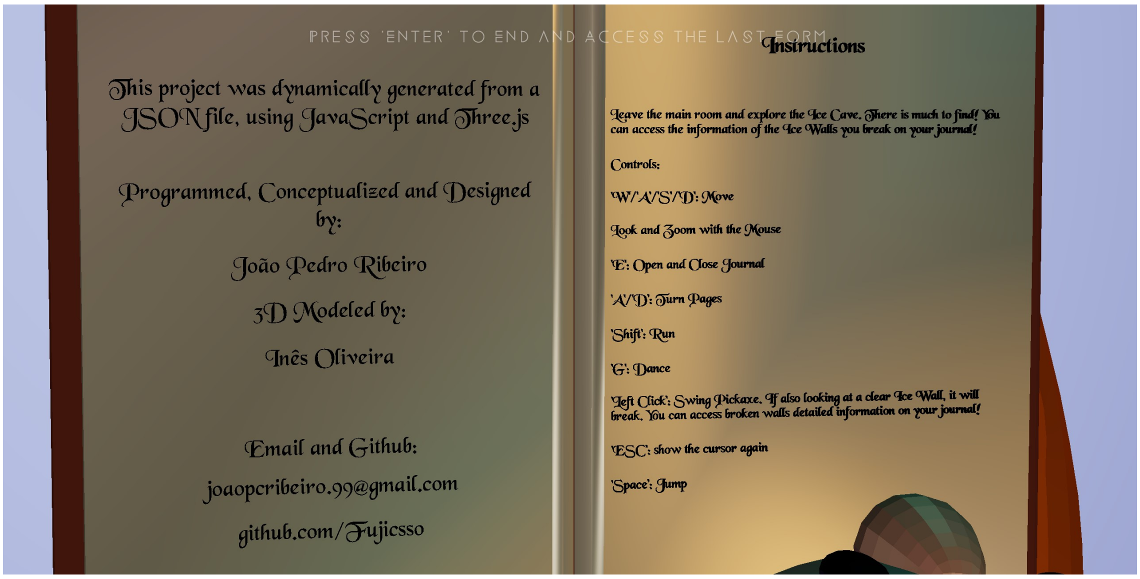


PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

'E' TO READ YOUR JOURNAL

Figure 4.2: Start Area

In this first section, the player is prompted on the bottom of the screen to open his journal (see Figure 4.2), where he can find the credits and basic instructions to the game (see Fig. 4.3) as this is the singular available page (for now).



PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

This project was dynamically generated from a JSON file, using JavaScript and Three.js

Programmed, Conceptualized and Designed by:

João Pedro Ribeiro

3D Modeled by:

Inês Oliveira

Email and Github:

joaopcribeiro.99@gmail.com

github.com/Fujicsso

Instructions

Leave the main room and explore the Ice Cave. There is much to find! You can access the information of the Ice Walls you break on your journal!

Controls:

'W'/'A'/'S'/'D': Move

Look and Zoom with the Mouse

'E': Open and Close Journal

'A'/'D': Turn Pages

'Shift': Run

'G': Dance

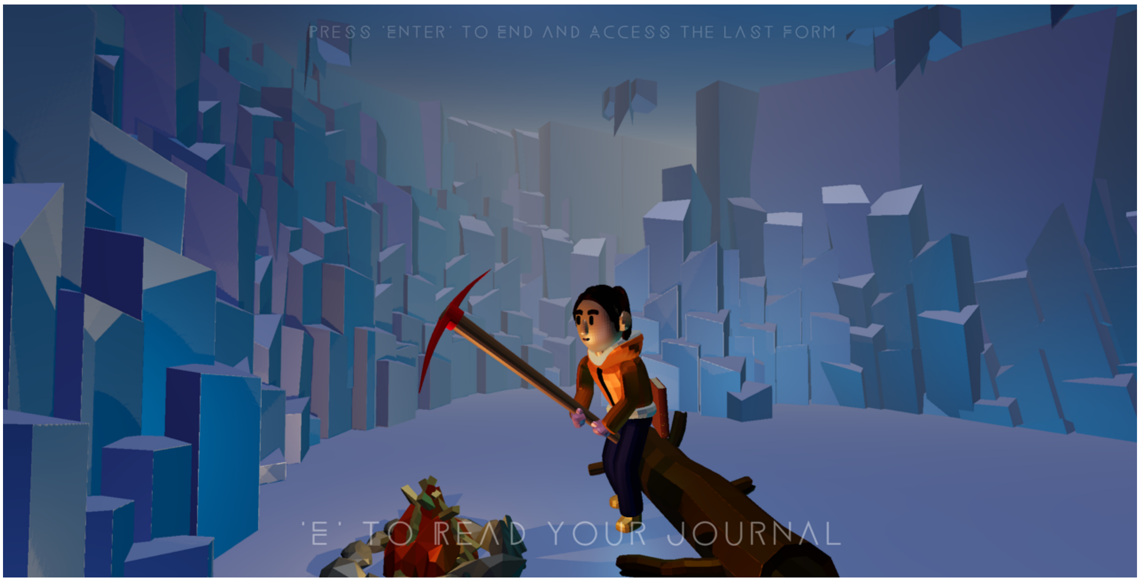
'Left Click': Swing Pickaxe. If also looking at a clear Ice Wall, it will break. You can access broken walls detailed information on your journal!

'ESC': show the cursor again

'Space': Jump

Figure 4.3: Credits and Instructions

The user now can close the book and test the controls and, when comfortable, they may now go through the narrow tunnel (see Figure 4.4) that connects to the next section. This separation was done with 2 things in mind. Firstly, to give the user less visual clutter right away and give a safe environment to try the controls. Secondly, as the user is not given a specific direction, it poses a very small challenge, impelling the user to look around and engage with the application. It's important for this first interaction to be extremely simple, as it could deter the lesser experienced.

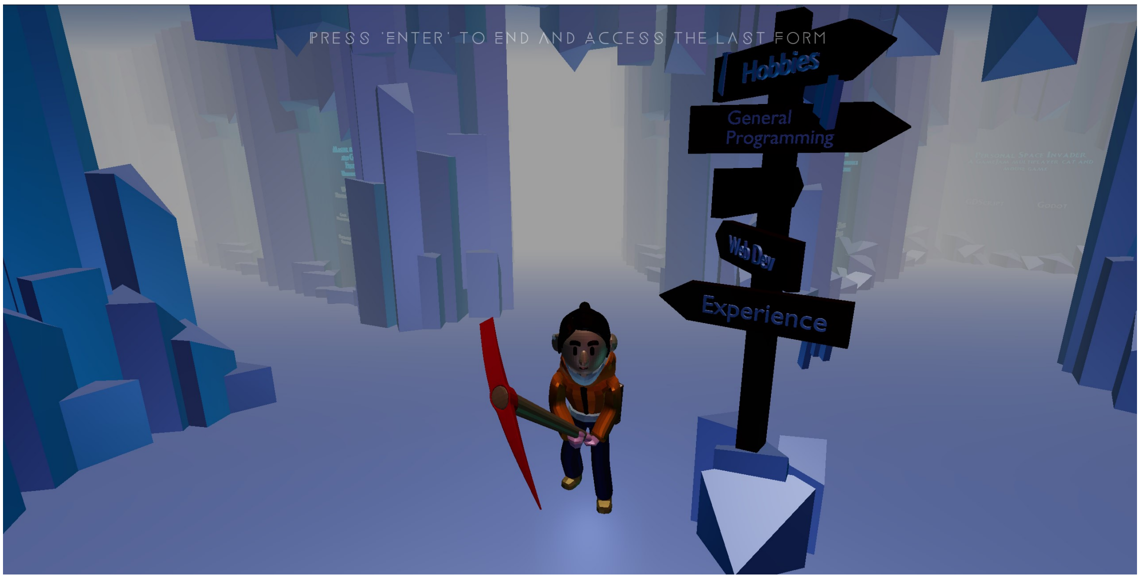


PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

'E' TO READ YOUR JOURNAL

Figure 4.4: Start Area - Narrow Tunnel

After traversing the narrow tunnel, the player arrives at the main area, which is the second section and where the game loop resides. Here the player has several wood signs that name and point each of the tunnels (see



PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

Figure 4.5: Main Area - Tunnels and Signs

In each tunnel, there is all projects, experiences and/or education regarding my CV. The criteria can be any as the website rearranges and generates the tunnels from a JSON file. As such, each tunnel has a theme, organizing and making the information more accessible.

Each tunnel divides and displays each individual content in a transparent ice wall, that displays the title, subtitle and key words associated with it. The keywords, for example, can be areas of study for education and tech used for projects (see Figure 4.6).

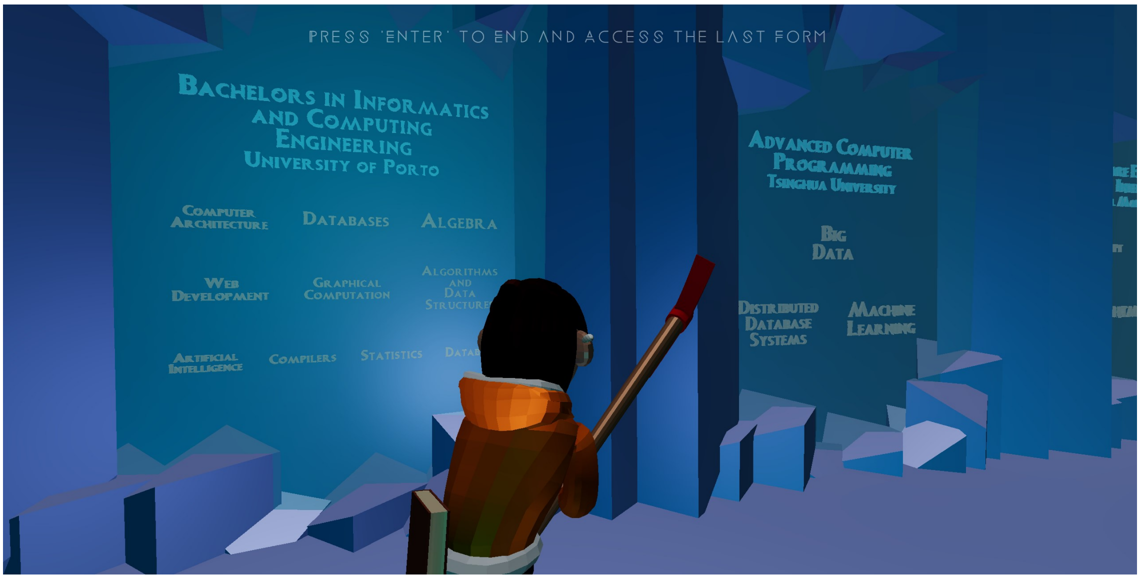
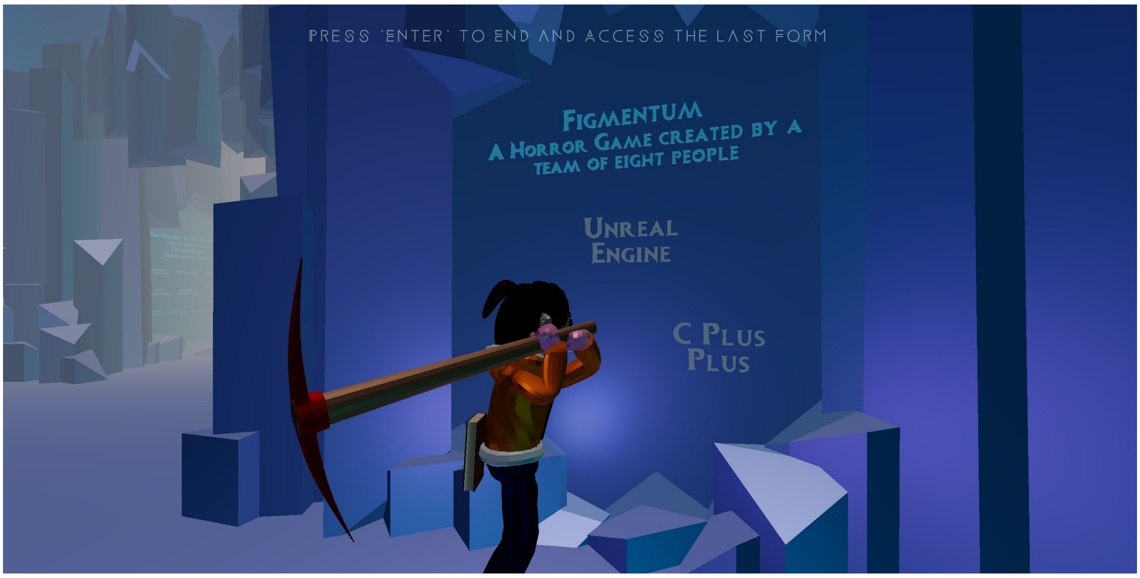

PRESS 'ENTER' TO END AND ACCESS THE LAST FORM
Bachelors in Informatics and Computing Engineering
University of Porto
Computer Architecture
Databases
Algebra
Web Development
Graphical Computation
Algorithms and Data Structures
Artificial Intelligence
Compilers
Statistics
Advanced Computer Programming
Tsinghua University
Big Data
Distributed Database Systems
Machine Learning

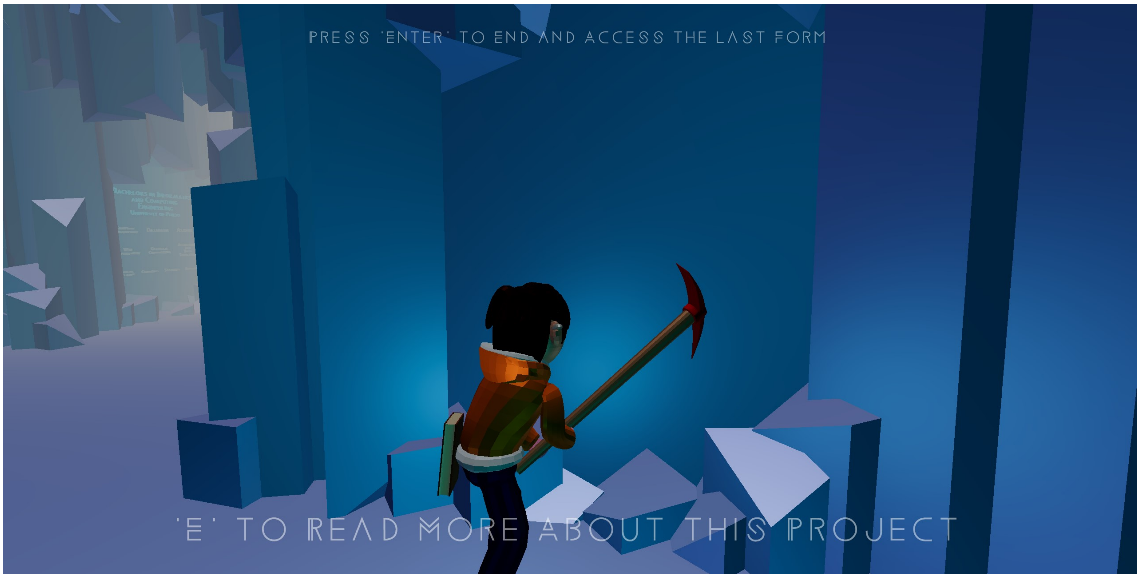
Figure 4.6: Tunnel

After the player finds some content they wish to know more about, by swinging the pick-axe while standing next to and looking towards the respective ice wall, it breaks (see Figure 4.7). Then the user is prompted to open the journal, as it has been updated with the new information (see Figure 4.8).



PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

Figure 4.7: Player Swinging Pick-Axe

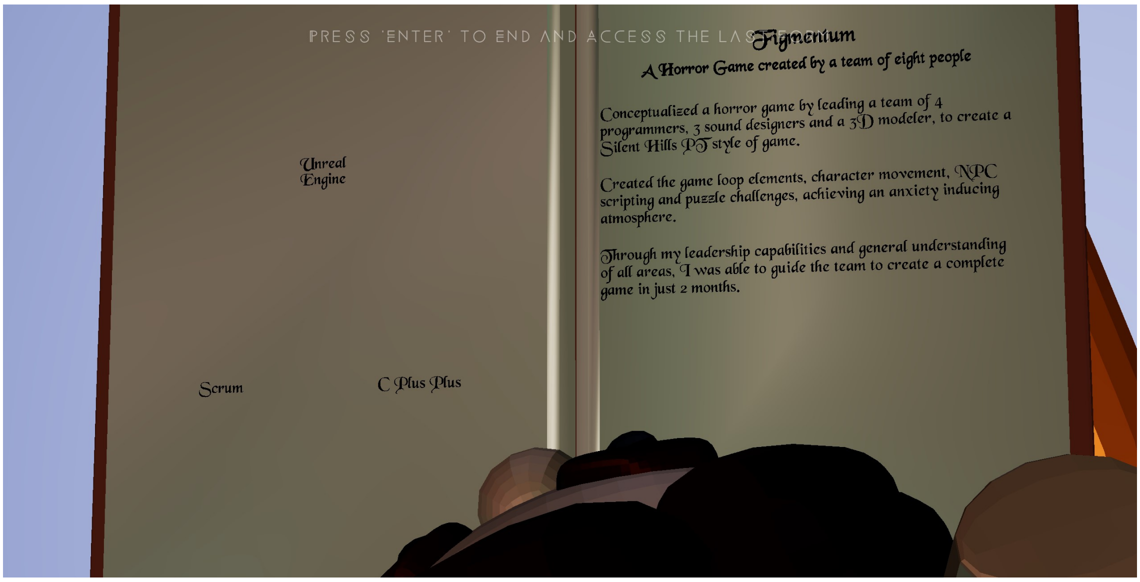


PRESS 'ENTER' TO END AND ACCESS THE LAST FORM

'E' TO READ MORE ABOUT THIS PROJECT

Figure 4.8: Prompt After the Ice Wall Breaking

When opening the book, the player is greeted with the last page opened but he can freely turn it right or left. The detailed information is placed according to the order of discovery, so the book is firstly divided and organized by section discovered and, inside it, ordered by specific content discovery. As for what information is displayed, it displays the same keywords as the ice wall on the left side of the book (these can be switched for images or 3D models) and the title, subtitle and full description on the right side (see Figure 4.9).



Figmentium

A Horror Game created by a team of eight people

Conceptualized a horror game by leading a team of 4 programmers, 3 sound designers and a 3D modeler, to create a Silent Hills PT style of game.

Created the game loop elements, character movement, NPC scripting and puzzle challenges, achieving an anxiety inducing atmosphere.

Through my leadership capabilities and general understanding of all areas, I was able to guide the team to create a complete game in just 2 months.

Figure 4.9: Book Detailed Information

Finally, the if the player pays attention, on the instructions page there is a special *easter egg* as there is a key for the character to dance to a song (see Figure 4.10).

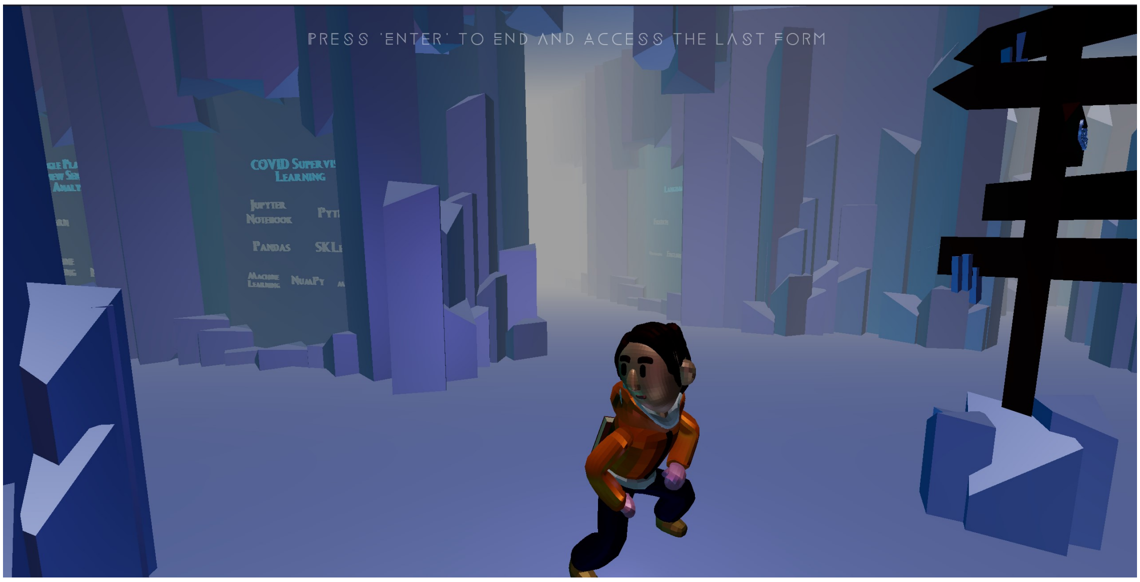


Figure 4.10: Character Dancing

4.4.1 Game Loop

As previously stated, the game loop is present on the second section of the game. Its the loop of exploring, breaking walls and reading on the book. While a simple loop, it is quite effective at engaging and propelling interaction with the user.

As this website needs to present information clearly and easily, the game loop has to be simple for anyone that is short time, but be enjoyable enough to impel users to spend more time and read more. As the book works as an inventory system, the user is incentivized to collect and explore.

By putting a wall between the summarized information and the full information, it forces a very small interaction that, by itself, makes the user more invested in its content. But, as mentioned in the Mechanics subsection, the obstacle must be minor, as too many steps to access the information would be a deterrent. This way, with the ice wall breaking and book opening, only 2 actions are required.

4.4.2 Player Controls

Controls and the games reaction to input are the key for a games ease of navigation and sense of spacial awareness. As such, the controls opted were in line with industry standards for third-person games. They are present at the first page of the characters book as seen on Figure 4.3.

For movement, the usual *W / A / S / D* serves to walk in all 4 directions. By pressing *Shift* the player is then able to run. *Space* allows jumping, but this has no mechanical use. It exists only for the sake of consistency.

Moving the *Mouse* changes the camera position to look around and using the *Mouse Wheel* zooms it in and out.

Pressing *E* opens the journal. Pressing it again while open closes the journal as well. While open, the user can move the *Mouse* to rotate the characters head and tilt the book. To turn the pages, *A / D* turns left and right, respectively. Alternatively the user can press *Left Click / Right Click* to also turn the pages, but this is not enabled on all browsers.

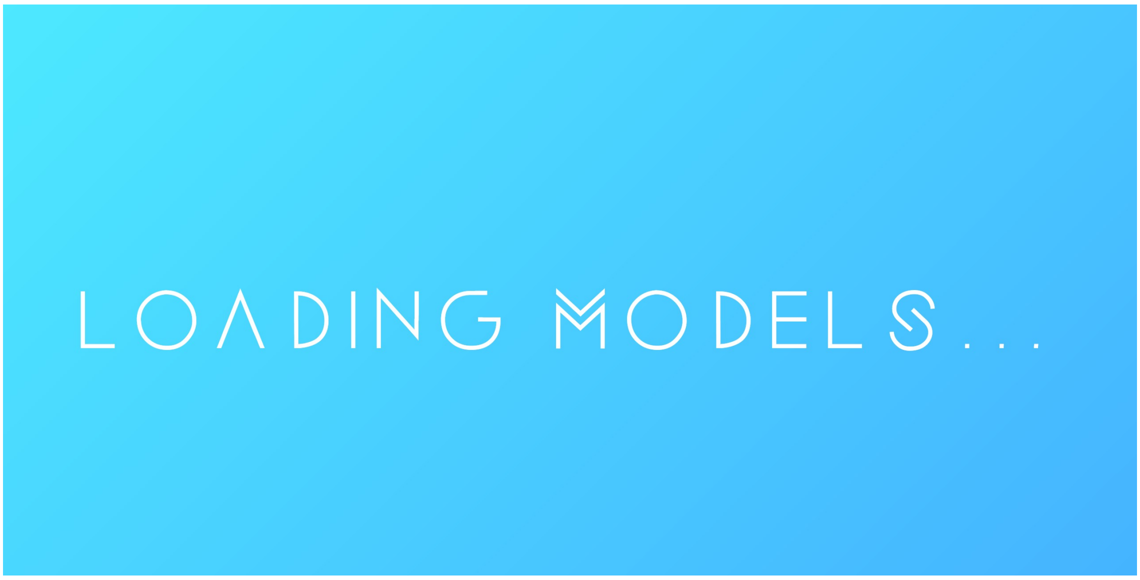
G Makes the character dance.

Finally, *Left Click*, while not having the book open to swing the pick-axe. Doing so while near a transparent ice wall and while looking at it will break it and update the information on the journal.

4.4.3 Screens

4.4.3.1 Loading Screen

The first thing the user sees when entering the website is an animated loading screen, as the loading of the models and processing takes a few seconds depending on the user's hardware Figure 4.11.



LOADING MODELS...

Figure 4.11: Loading Screen

4.4.3.2 Start Screen

After the website finishes loading and setting up the scene, the user is greeted with a panoramic view of the player sitting on a log next to a campfire, seen from the hole where the player fell through (see Figure 4.1). Upon left clicking the screen, the global ice cave audio starts playing, as well as the positional campfire audio. This way the player starts the immersion even before taking control of the character.

4.4.3.3 Character Screen

This is the most important screen of the game, where the player can move around the environment and interact with it (see Figure 4.5). All character animations, movement and camera movement is present here. Also, the audio for walking, running and breaking ice walls cue according to the users inputs and interactions.

4.4.3.4 Book Screen

With the objective of reading either the instructions and credits or the detailed information of the broken walls, the player opens up his journal. Here, beyond turning pages, the user can tilt the book, turn the character's head and zoom in/out to better read and increase immersion, as if the player was holding the book himself (see Figure 4.9).

4.5 Resources

4.5.1 3D Assets

For a 3D game, its assets are the key to the visual setting and design. As such, while this is a thesis on the programming of the game, this cannot be overlooked as its weight on the whole experience is too great.

Some 3D models were modeled from scratch according to my vision by the artist Inês Oliveira, such as the player character, the opaque ice walls and the signs. The character was modeled after myself, as the website details my CV. The map itself was planned and designed by me, from which she modeled the main areas and tunnel.

The models used in this game are: character, closed book, open book, pick-axe, main map, tunnel, tunnel end with odd ice walls, tunnel end with even ice walls, campfire, log and signs. The tunnels are cloned, sorted and arranged according to the information JSON. All other visible models, such as the text on the book and ice walls with the respective text, were dynamically generated from the information JSON.

4.5.2 Animations

As animations are too time consuming for the scope of this project, they were all generated using Mixamo, a free Adobe owned online repository that generates FBX animation files for an unploaded mesh. It also dealt with the skinning of the mesh (process where a skeleton is generated and linked to the mesh). Some time-consuming tweaks were still necessary, as each equipable item had to be individually skinned and added to the main skeleton, as the Mixamo skinning process does not take solid items into account.

There is a total of 18 animations, each associated with its respective character state.

4.5.3 Audio

The audio was all taken from free repositories of audio sound effects and samples. Each was picked and adapted to match with the animations.

The audio used was an ambience ice temple music and sound effects for walking, running, breaking ice and fire crackling. While more audio could be added, this was proven plenty and more audio would make the loading times longer with diminishing returns on user engagement.

4.5.4 Thee.js

The framework used was three.js, a veteran in the 3D Web space, having its first iteration launched in 2010. It uses WebGL like most to render and handle everything under the hood.

Using JavaScript one can program anything using Vectors, Quaternions, Models and Cameras. While there are already many officially approved add-ons already, there's still many missing for the usage on 3D games. While there technically is an engine called *Rogue Engine*, it's still very limited and it's more productive to use vanilla three.js and develop tools for it.

The only add-ons used were the FBX Loaders and SVG Loaders, all used to turn these files into three.js objects (akin to JSON files).

4.6 Summary

To summarize, this game allows the user to immerse into a carefully curated personalized experience that only requires text update to display more projects as they need to be added to the CV. It streamlines the future maintenance for the maker, while having many features that help with future features as it will be described next.

This experience, through the use of many asset types, creates an environment that helps reel in the user and engage with the content being displayed. One is pushed to explore everything but may just go to a specific section of interest for efficiency sake. While slower than a typical 2D website, the difference is mostly negligible with the major upside of pushing for spending more time and reading more.

Chapter 5

Implementation and Tools Developed

This chapter is dedicated to the structure, architecture, logic, implementation of the project and, in the end, the tools developed for the three.js community.

5.1 File Structure

As this is a static page, the code consists only of JavaScript, CSS and HTML files. There are also some JSX files as React was used to organize the page rendering.

The structure is the following:

- `public` - contains all the assets, fonts, audio and *SVG*s used:
  - `assets` - this folder is subdivided for each asset type: `environment`, `map`, `player character`, `tools` and `sound`.
  - `fonts` - this folder has all the font files used, in the *TTF* and *JSON* formats. The title and prompts use *TTF* while the 3D text generation requires *JSON*.
  - `svgs` - this folder houses all the *SVG*s related to the tech used on the presented content. While they ended up not being used in the final version, the tool that generates 3D models with relief was still developed.
- `src` - contains all the *JSON*, *CSS*, *JSX* and JavaScript files used:
  - `lib` - this is the most important folder, as it has all the JavaScript files, responsible from the loading, setup, generation and rendering of the whole game: `Book`, `Camera`, `Map`, `SceneInit` and `SvgImporter`.
  - `json` - this folder contains all the information (*CV* content and Tech *SVG* paths) files with the *JSON* format. This format was chosen to facilitate the implementation into the game generation while being readable and easily altered at any time with no code change.

  - `jsx` and `css` - These files, directly under the `src` file, are responsible for mounting the React content into *HTML* (this includes both forms, the rendering canvas and text for the prompts and title) and respective styling and animations.

- `package` - typical *npm* package file with everything needed to be installed to run the project.

- `config` - typical Vite package file with everything needed to be installed to run the project.

5.2 Game Architecture

The game logic resides entirely on the `lib` folder, being separated into 6 JavaScript Files. In this section I will go over each file after explaining the overarching structure.

5.2.1 Approach

An Object Oriented approach was used, focusing on the use of classes over functions. This way it guarantees that these classes can be modularized and then re-used like a framework.

While everything was written on JavaScript, three.js actually is compatible with Typescript. While this language is growing rapidly, I chose JavaScript as TypeScript is backwards compatible with it.

5.2.2 Canvas

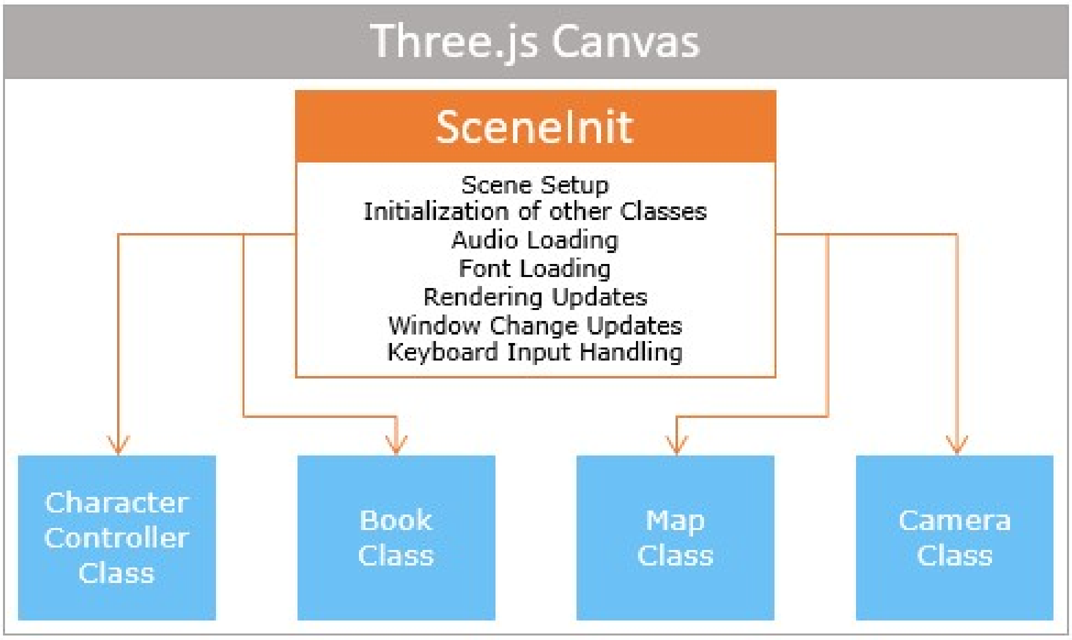


Figure 5.1: Code Structure

The canvas is the **HTML** element that three.js uses to draw onto the web browser. Thus, this canvas must be updated through a method in JavaScript. In this case, the `SceneInit` class is responsible for the initial setup and constant update of the canvas element, while the other classes handle the actualization of the three.js scene according to inputs, interactions and time (see Figure 5.1).

5.2.3 Classes

Each **JS** file represents a main class, together with its helper functions and helper classes. They all serve the common purpose of inserting or updating objects within the three.js scene. While they are modular in theory, most of them need updates from each other, as many actions depend on different areas of the game logic. In Figure 5.1 we can see that each class is co-dependent of one another, but this was merely the approach chosen out of simplicity of implementation. The `SceneInit` class could update each class information accordingly, removing any dependence. It's up to the programmer how to use these classes as they will be made available to the public.

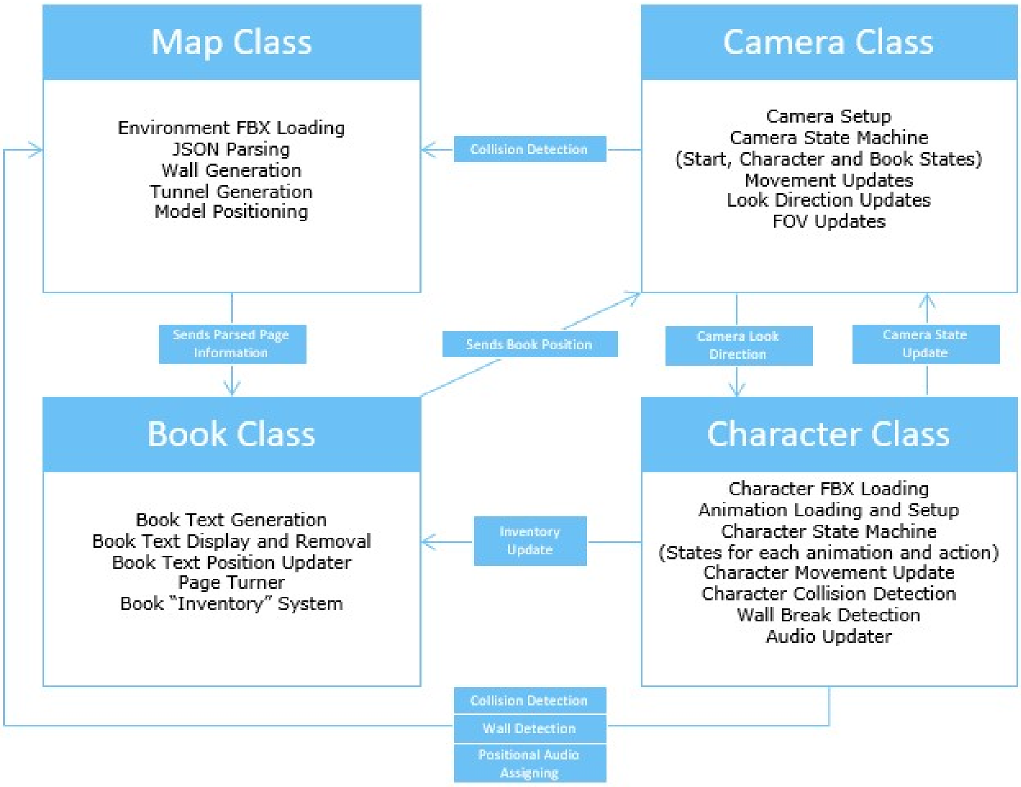


Figure 5.2: Class Structure

Each **JS** file represents a main class, together with its helper functions and helper classes. Each of them tackles a different area of the game, and each are explained in the following subsections.

A visual summarized explanation can be seen on Figure 5.2 All classes have their respective constructor method but all they do is instantiate the passed parameters into class variables and set all variables to be used to their initial value (if there is no initial value, they will be set as undefined). As such I will not go through each classes constructor.

5.2.3.1 SceneInit

This class has the following methods:

- **_Initialize**

  This is the main method of this class, as it initializes everything in the game.

  Firstly, it creates the scene and canvas.

  Then it parses the JSON containing the content into a two dimensional array, where each array inside the main array has all contents of the same type. At the same time, it also creates an array containing each type name. All this information is organized this way for the later tunnel, ice wall and text generation.

  Afterwards it sets the renderer up. Here many performance related settings may be used, including the type of renderer. WebGL was the renderer chosen for its wide use and extensive documentation.

  Audio is the next to be initialized. The audio loading is done before anything else as it affects several classes and, this way, when they are started they already have these variables at hand.

  Next is the `Map` setup. A new instance is created and then initialized. The sound for the campfire, which is positional, is passed to the map to be added to the campfire scene object.

  Then the **LoadAnimatedModel** method is called and its class variable is passed onto the `Camera` and `Map` classes.

  Event listeners for window resizing, keyboard press and keyboard release are added, calling the **onWindowResize**, **onKeyDown** and **onKeyUp** respectively when activated.

  Finally, the **RAF** method is called.

  This all completes the initial setup, to be finished inside each of the other classes.

- **LoadAnimatedModel**

  This method only sets up the parameters and sounds and creates a new `BasicCharacterController` class.

- **RAF**

  This is a recursive method responsible for each frame update on the canvas. It requests and tracks the animation frame, which in turn will allow for the tracking of frames between updates for better character and camera movement and animations. Otherwise, slower computers would see everything in slow motion.

It also calls the `Map` **UpdateLights** method, the **Render** method and the **Step** method.

- **Step**

  This method takes the time elapsed between calls as a parameter, calculated by the **RAF** method.

  It then calls all frame sensitive updates from the animation variable, the `Character Controler` updater and `Camera` updater.

- **Render**

  All this method does is call the renderer to render the scene according to the current camera.

- **onKeyDown**

  This method is called on the press of a keyboard key. It then stores the respective boolean for other classes to use.

- **onKeyUp**

  It does the same as the **onKeyDown** except it triggers when releasing the key, setting the correspondent boolean to false.

- **onWindowResize**

  The method is triggered when the window resizes, to update the cameras rendering frustum near and far plane aspect ratio.

5.2.3.2 Map

This file not only has the `Map` class, but it also has some help functions.

A **chunkify** function to aggregate an array into a width first even bi-dimensional array. This helps with the positioning of the tech information on the wall and journal.

A **wrapText3D** function developed to simulate the wrapping capabilities that HTML and all text editors have, but on 3D text. It takes the text, max width, line spacing and text alignment as parameters. It uses three.js's TextGeometry function to generate the 3D text according to a given font.

And finally a **AddWallContent** funtion that returns an object containing a transparent ice wall, and the title, subtitle and tech matrix generated wrapped text at a given position. This wall wrapps each individual text according to a subdivided grid set relative to the walls total dimensions. This means all text font is resized as needed and re-wrapped along its designated 3D section. This way, any text on any size of wall is generated flawlessly. All models can be customized dynamically too by passing the geometry and material parameters.

The `Map` class has the following methods:

- **MapInit**

  The main purpose is to load all the **FBX**s related to the map and environment and adding them to the scene. After that, the scenes 11 lights of cold hues are created, consisting of an ambient dim light, 4 locked directional lights at different angles to give different hue reflections to all sides of the wall's icicles, 3 rotating directional lights to simulate the ice refraction effects, and 3 slow intermittent point lights that accompany the player to make the ice change its hue according to the movement and proximity of the character too. It also adds one warm unstable point light to simulate the campfire's fire light.

  It also adds the non tunnels to the collision array that the character then utilizes for the movement calculation.

- **LoadTunnels**

  This function is one of the slowest ones, as it takes the content and generates a tunnel group for each section. From that section, according to the parity and number of walls, it will generate one end tunnel clone (even or odd version) and all open-ended tunnel clones, all positioned sequentially. Then generates the content walls for each section and position them. Finally each tunnel group is rotated and positioned in its respective entrance and the correspondent section type text is added to the directions sign.

  This function also has an non-utilized developed function that returns a random angle within a range at specific odds to make each section of the tunnel have a randomized but controlled curve to it to appear more natural.

  It then adds the tunnel groups to the collision array that the character then utilizes for the movement calculation. The ice walls are also added to a separate array to detect if the camera is looking towards one.

  At the end, all objects generated are added to the scene.

- **UpdateLights**

  This function just updates the moving lights: updates the rotation of the rotating directional lights, the position and intensity of the intermittent point lights and the intensity of the campfire light.

5.2.3.3 Camera

This file actually has several classes, most dedicated to setting up a State Machine to handle the different behaviours the camera must have across the game.

As three.js had very limited pre-made cameras, this camera was developed from scratch to enable mouse capture on a third person camera, and apply smoothing interpolations on the camera position, "look at" position, orbit point position and distance to the later.

Firstly, the main one is the `Cam` class composed of the following methods:

- **CamInit**

  This method initializes the camera, raycaster, mouse event listeners and pointer locking. The raycaster is used to detect collisions with the map and adjust the camera distance to the orbit point accordingly.

- **LockChangeCallBack**

  This method detects when the point is locked, to allow for camera and character movement, and cue the ambience and fire sounds.

- **MouseMoveCallback**

  This method updates the rotation variables according the the mouse relative X and Y changes, depending on the current Camera State. These rotation variables affect the camera current position, look at, up vector and character head rotation. These all contribute for the camera to have smooth and responsive movement.

- **MouseWheelCallback**

  This method updates the distance the camera is from the orbit point. This has different functions depending if the camera is on the Character State or Book State.

- **Update**

  This method calls the **Update** method of the current state, which will be explained next.

The rest of the classes are grandchildren of the grandparent class `FiniteStateMachine`. This class acts as an interface with the methods **AddState**, **SetState** and **Update**, and variables for all the states, current state and `Camera` Class scope.

AddState adds a new state to the **states** object.

SetState calls the previous state **Exit** method, sets the current state variable and calls the new state's **Enter** method.

Update simply calls the **Update Input** and **Update Cam** methods of the current state.

`CameraFSM` extends `FiniteStateMachine` adding a new method **Init** to add all desired Camera State Classes to the State Machine.

All that is left are the Camera State Classes. They all extend from the same State Class, with **Enter**, **Exit**, **Update Cam** and **Update Input** methods. All do not update themselves as per my choice as I wanted to link them to the Character States, but this is not mandatory by design, as the **Update Input** method exists to change the current Camera State according to any designated input.

- `Start State`

  This state only implements the **Update Cam** method, keeping it at a locked position and "look at" position.

- `Character State`

  The **Enter** method sets a class boolean (**can_update**) to false that stops a section of **Update Cam** method for a timeout, ensuring a sequenced movement on entering this state. This works differently depending on the previous state.

  The **Update Cam** method is quite complex, as the whole camera logic was developed and implemented here. Depending on the previous state, a time variable is calculated according to the time elapsed from the last frame and a given number, which will be used to set the vector lerping (interpolation between 2 vector values) duration. Then, a raycast of limited length positioned at the center of the camera, aligned with its direction, is calculated to check for map collisions with the camera and, if positive, changes the camera distance to orbit point variable to the current value minus the intersection position.

  Then, if the cameras **can_update** is true, it lerps the orbit point according to the character direction. This way the orbit point follows the same relative position to the character in a smooth manner.

  Finally, with some trigonometry the cameras position, "look at" and orientation are calculated from the orbit point, distance and rotation variables previously set.

- `Book State`

  The **Enter** method fetches the Head's Bone from the Character Skinned Mesh and gets its world direction variable (world direction is absolute, local direction is relative to parents), resets the current book mouse position and sets 2 timeouts: one to allow the cameras up vector update and one to set the camera in place while in this state.

  The **Update Cam** method does the same as the Character State's **Update Cam** method, with the addition of getting the Open Book's Skinned Mesh and its Bone. Using it's position and world direction, the camera can then go through a sequence: first it lerps its position towards a set height from the book's bone position and the "look at" towards the open book's bone position. After the camera is able to update is up vector, it lerps it to the skinned mesh direction, matching both and aligning the camera to the books orientation.

  Only then, when the camera reaches its fixed position, it then can lerp ever so slightly the position and "look at" to simulate a book tilting effect.

  This method also updates the Head's Bone rotation according to the mouse input.

  The **Exit** method just sets the Head's Bone rotation back to the value before entering this State, to realign it.

- `Wall State`

  This class ended up not being used and fully developed as it was not worth it for the scope of the project. It would simply cause a semi-lock-on on a wall when looking at it to have a better POV to improve usability, akin to the Book State. As the third-person camera proved to be good in that category, the focus was diverted to other more important features.

5.2.3.4 Character Controller

This is, by far, the most complicated file as it houses several individual classes, the State Machine's Classes and some helper functions.

The first class is actually a duo of classes, as `BasicCharacterControllerProxy` only serves as a proxy for accessing and storing the `BasicCharacterController`'s animations variable.

The `BasicCharacterController` has the following methods:

- **Init** Firstly, many class variables are defined here, such as: movement related variables (acceleration, decceleration, velocity and position), animation storage and handler variables, and form related variables. Then, the **LoadModels** method is called, as it will work in asynchronously. The clock for the storage of time spent on the website is initialized here. Then, some event listeners are created related to both forms: one tracks the Start Forms loads, keeping a counter to change the z-index of the HTML element (to hide it) and start the clock timer, the other tracks when the End Form loads, keeping a counter to submit the hidden form (responsible for sending the clock's value to an external google sheet) when the user submits the End Form. This is needed as the submission of the hidden form forces a redirect, which resets the page.

- **onMouseClick**

  This method, activated on a mouse press through an event listener, checks if the character is currently walking or idle. Then, if yes, sets it's current state is set to the **Swing** State. Then it sets a raycaster from the center of the camera in the same direction as the camera, checking for intersections with the content ice walls. If it detects one within a threshold distance, it then checks how far the character's position is to that wall. If it is near enough, then the wall is stored in a class variable to be then altered at the end of the swing animation, the content associated to that wall is passed down to the `Book`'s inventory array and it sets timeouts to fade in the prompts to read the journal to access the new information using **CSS** animations.

- **onKeyDown**

  This method merely checks when the user presses **Enter** to change the z-index of the End form (to hide it) and sets the hidden forms value to the clock timer value.

- **LoadModels**

  Firstly, it loads the characters skinned mesh using an FBX Loader. This is important as all other FBX are unskinned, containing only the bone animations.

  After loading, it will be parsed to separate key meshes and bones that will be used to align the camera, toggle visibility, adding positional sound and turning the head. The camera alignment and head turning was already covered previously on the Camera Class.

The visibility toggle is important as the skinned mesh contains uni-boned skinned meshes for each equipment in each "fixed" position. This means there's a closed book on the characters waist, an open book on the characters left hand, a pick-axe on the characters right hand and a pick axe on the characters back. Each must be toggled perfectly between animations.

Then, it loads all animations and, for every one of them, adds them to the animation array.

Finally, after everything is loaded, the **onCharacterLoad** method is called.

Then the

- **onCharacterLoad**

  In the first place, this method slows or speeds up the walking and running sounds according to their duration compared to the animation's duration. This ensures they are synced and proceeds to then add them to the character's right boot mesh (the sound is positional, so this helps with the immersion).

  Afterwards, the Character State Machine is set to the `Sitting Idle` state, which is the starting state for the character, and it is then added to the scene.

  Then, the `Book` Class is initiated and a function to be run after the `Book` is finished loading and generating its assets. This function serves to remove the loading screen and, if when loaded there is no form, to queue the main title, subtile and prompt to start to fade in with specific timeouts as to create a cinematic experience.

- **Update**

  This method is summarized into mathematics to calculate the movement and rotations of the main character and updating the animations and Character State. These only happen if the mouse is locked, to stop some unwanted interactions.

  First, going over the movement and rotation calculation, as they are tied together:

  - **Movement**: The character's movement is calculated by reading the input from the **W A S D** keys, meaning forward, left, backwards and right respectively. The X movement vector's polarity is positive when going forward, while the Z movement vector's polarity is positive when going to the right.

    If there is no input, the current velocity (different for the X and Z vectors) is taken and multiplied by the set deccelaration constant and the result is squared. This creates a smooth deccelaration.

    If there is input, the method checks if it is running to double the acceleration values, else it checks if the character isnt either walking or jumping to set the accelaration to zero, which keeps the character in place and not rotating.

    Depending on the inputs, the new velocity is calculated taking into account the time passed since the last frame, which guarantees a decent experience even is the user's computer is cant run this website well.

The new velocity then is lerped into the characters current velocity with a set time delay.

- **Rotation**

  To ensure the character is rotated relative to the movement and the cameras direction, this function takes the cameras quaternion, nulifies the Y value, and applies it to a positive X unit vector (forward vector) and a positive Z unit vector (sideways vector). It then multiplies each by the current velocity vector's X and Z parameters respectively. Finally both are merged as a 2 dimentional vector (X and Z) and a single movement vector adjusted for the rotation is created. The characters rotation is also adjusted using this vectors angle to a positive unit X vector and rotating the character via Y axis by that amount, with a small lerp to smooth it out.

- **Collisions**

  This was kept to the basics, as there is no need for complex collision systems on a already demanding application that is only exploration based with simple movement mechanics. As such, a single raycaster set from the characters movement vector direction is set at knee height. If it doesn't intersect with any of the map meshes within a threshold, the movement and rotation are applied to the character's scene object.

- **State and Animations**

  Finally this method updates both the Character State, passing the time passed since the last frame and input status as parameters, and the animation mixer, which handles animations similarly to a State Machine (it is already built into three.js, so there was no need to develop).

As for the Character State Machine, it functions the same way as the Camera State Machine as it extends the same `FiniteStateMachine` Class. This class acts as an interface with the methods **AddState**, **SetState** and **Update**, and variables for all the states, current state and `Character` Class scope.

`CharacterFSM` extends `FiniteStateMachine` adding a new method **Init** to add all desired Character State Classes to the State Machine.

There are a total of 18 Character States, each with an assigned animation, being divided into 2 types: Loopable and Single Loop animations (see Figure 5.3). Single Loop animation states, represented with the Blue Rectangle, either stop after the loop or default to another state (usually the previous state, seen with the yellow arrow). The Loopable animation states, represented with the Yellow Rectangle, either loop while there's no input or default to another state if a specific input is released. Red Arrows symbolize State change from input update. Yellow arrows symbolize automatic State change as mentioned previously.

All Single Loop Animation States include an extra *finished* event listener and both a **Finished** and **Cleanup**, to end and update the state on animation end. To smooth transitions out, this method only updates a class boolean that then enables the **Update** method to work as any of the Loopable Animation States, defaulting to idle or changing if the user, by chance, is pressing a key. For example, while unsheathing, if the user is pressing W, he will change the state to the `Walk` State on the end of the current state, otherwise it would force the state to be `Idle` State before changing to the `Walk` State. A way to visualize this on the Diagram, is to think of yellow arrows pointing towards a State that changes on input (has outbound red arrows) as ending on wherever the red arrow corresponding to the input would end.

Loopable Animation States only update when there's an input update.

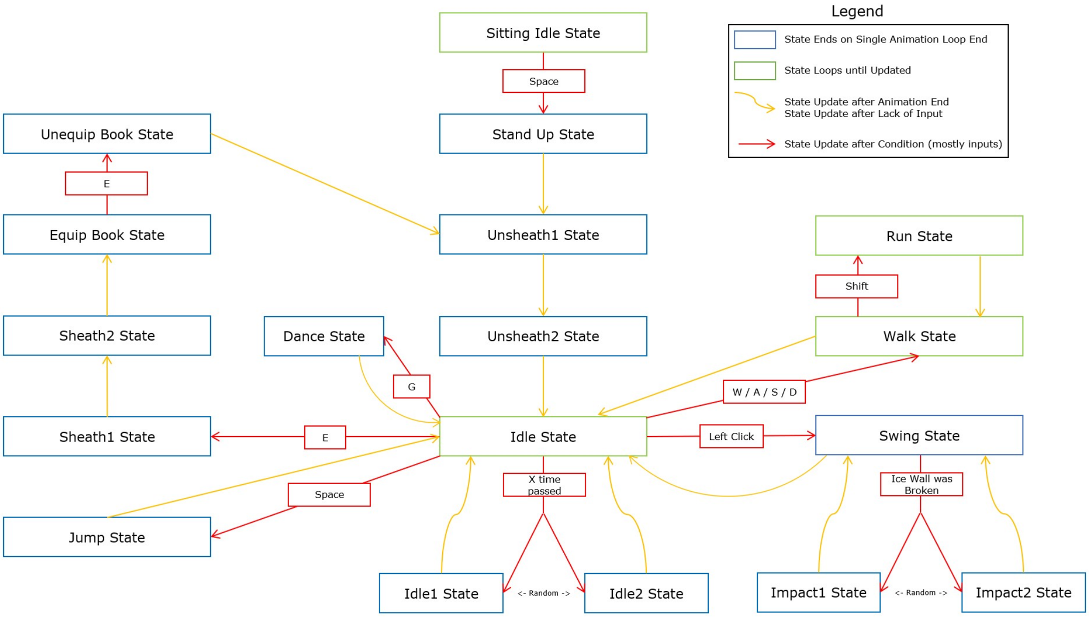


Figure 5.3: Character Class Relational Diagram

All of these classes have the same structure:

- **Enter** This method takes the previous state's animation (if it exists) and *cross fades* it with the current state, while matching the animation frame. This allows for the seamless transition between animations.
- **Update** This method sets the new state according to which input's are currently pressed. There is a hierarchical order given towards the states that should be prioritized.
- **Exit** Mostly just used to call the **Cleanup** method on Single Loop Animation States.

Having that said, a couple states have some important functionalities:

- `Sitting Idle State`

  On **Enter** sets the pick-axes' visibility, fades in the main title, subtitle and start prompt and sets the Camera's State to `Start`.

  On **Exit** fades out the main title, subtitle and start prompt, and fades in the permanent top end form prompt.

- `Stand Up State`

  On **Enter** sets the Camera's State to `Character`.

  On **Exit** fades in the initial prompt to open the journal.

- `Dance State`

  On **Enter** and **Exit** plays and stops a short music.

- `Walk State` and `Run State`

  On **Enter** and **Exit** plays and stops a short looped sound effect.

- `Sheath1 State`

  On **Enter**, if the initial prompt to open the journal is still visible, adds a fade out to it. Only then it adds a fade in for the post broken wall prompt.

  On **Enter** sets the Camera's State to `Book`.

  On **Finished** sets the pick-axes' visibility.

- `Sheath2 State`

  On **Finished** sets the books' visibility.

- `Equip Book State`

  On **Finished** calls the `Book`'s **DisplayCurrentPage3DText** method.

- `Unequip Book State`

  On **Enter** sets the Camera's State to `Character`.

  On **Enter** calls the `Book`'s **RemoveCurrentPage3DText** method.

  On **Finished** sets the books' visibility.

- `Unsheath1 State`

  On **Finished** sets the pick-axes' visibility.

- `Swing State`

  On **Finished**, it checks if the *intersected wall* variable is set. If so, it cues the Ice Breaking Sound and turns the respective wall and children invisible. After that, it randomly selects the next state as `Impact1` or `Impact2` .

5.2.3.5 Book

This class is dedicated to the loading, generation and display of the book's text, and the updating and navigation of the inventory system.

The inventory system starts as a blank 2D array, with only the credits and instructions page (see Figure 4.3 available at the first position of the first array. As the user breaks walls, the inventory first checks if its type/section is already present in any page on the inventory. If not, it adds the page to empty array. If yes, it adds the page to the end of the corresponding array. This way the user accesses the new content in a first served manner within a logical and organized section.

The `Book` class has the following methods:

- **BookInit**

  This method loads the font for the book text and, after loading, calls the **LoadAllPages** method. It also adds the event listeners for key press and mouse click: **onKeyDown** and **onMouseClick**

- **LoadPage**

  Akin to the **AddWallContent** funtion on the `Map` Class, this takes a page and generates the left and right side of the book, returning it as an object. On the left side is the same 2D matrix of tech as seen on the content walls. On the right side is the title, subtile and the full description with paragraphs. All use the same **wrapText3D** function to have the desired effect.

- **LoadAllPages**

  This function iterates through the full content array to call the **LoadPage** method on each of them, adding the result to an array of leaded pages. It also generates the instructions and credits page, but this one is not present on the JSON file as it is hard-coded.

- **ReturnNextPage**

  Due to the inventory structure, this helper method was created to find the next double index if the user wants to turn pages right or left.

- **AddPageToInventory**

  This method, called by the `Character` Class upon ice wall break,

- **DisplayCurrentPage3DText**

  This method gets a clone of the current page from the loaded pages, positions and aligns it with the book using its World Position (Book Bone) and World Direction (Book Mesh) and adds it to the scene.

- **RemoveCurrentPage3DText**

  This method searches the scene for the current displayed page and removes it.

- **onKeyDown**

  This method simply calls the methods **RemoveCurrentPage3DText**, **ReturnNextPage** and

- **DisplayCurrentPage3DText** in this order according to the input (A -> Left or D -> Right).

- **onMouseClick**

  This method is an exact copy of the **onKeyDown** method, expect it takes Left and Right Click as inputs.

5.2.3.6 SVG Importer and Modeler

Its purpose is to take an SVG file, convert it into a three.js shape and then iterate through each SVG path to extrude a mesh with the same color value. These meshes then would have different thicknesses (first path parsed has the lowest thickness) as the SVG image file works with layers. This way the final mesh has relief according to the layer height.

While this part is indeed finished, due to time constraints it was not possible to adapt it to SVGs with gradient colors and elements other than paths. As such I opted with the generic 3D text since, with some of the tech having logos with gradients, presenting some 3D models and some 3D text at random in the same sectioned area would not be cohesive.

5.3 Interface

The interface was already mentioned many times, as its only composed of the **Start Overlay** (Main title and subtitle) and all the prompts (**start prompt**, **initial book prompt**, **wall break book prompt** and **end form prompt**).

It was built only using an imported font, HTML and CSS styles and transitions for the fading animations. Every animation, while handled by CSS, was triggered by the JavaScript game classes.

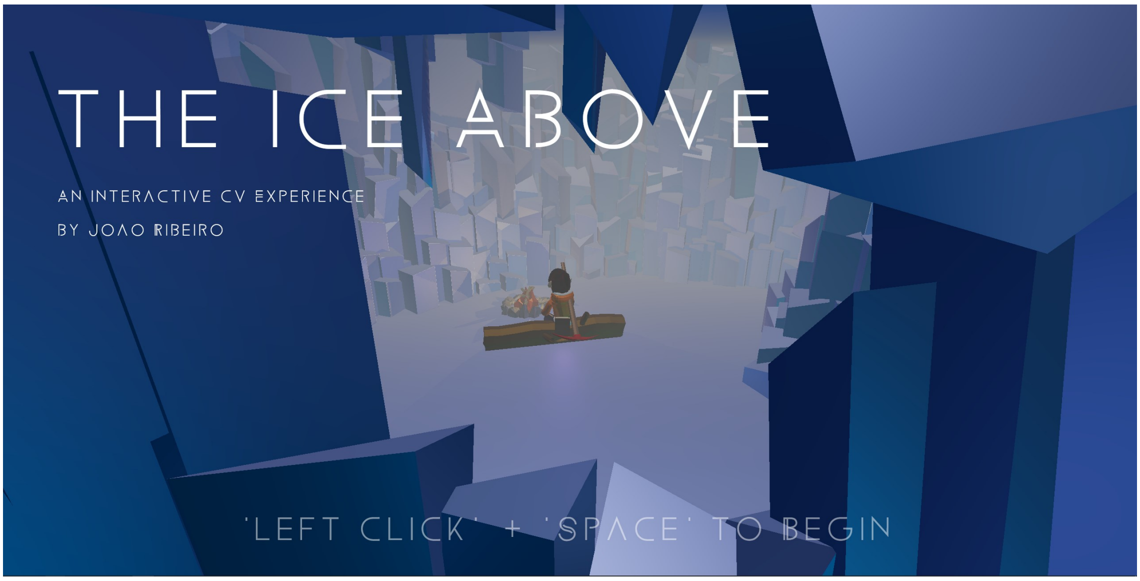


Figure 5.4: Main Screen with Start Prompt

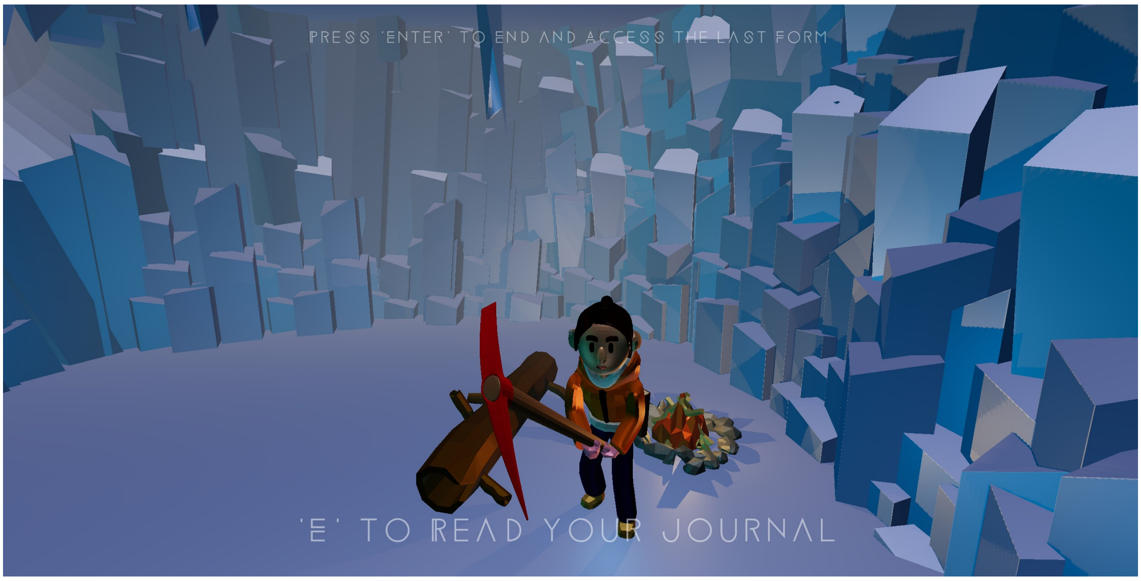


Figure 5.5: Initial Book Prompt and End Form Prompt

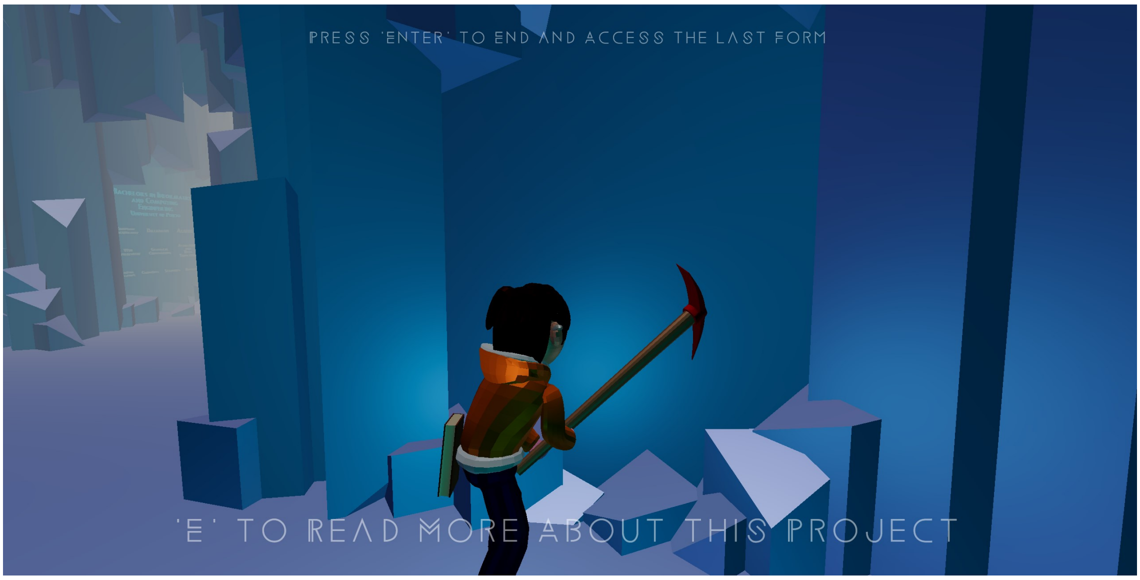

PRESS 'ENTER' TO END AND ACCESS THE LAST FORM
'E' TO READ MORE ABOUT THIS PROJECT

Figure 5.6: Wall Break Book Prompt

5.4 Forms

There are 2 google forms: one at the start and one at the end. Each was directly embedded on the HTML and it works as a separate entity hosted by a different server. This way I can use the google analytic tools to analyse and evaluate the responses.

There is also a hidden form, but it works as a metric gatherer.

5.5 Metrics

The most important metric to gather, regarding user engagement and retention, is the time spent on a website. To this end, a hidden HTML form with a single value was placed behind the three.js canvas.

This value is the time the three.js **Clock** counted since the first form was closed until the end form was openned. It is submited automatically at the same time as the End Form, because its submition forces a page redirection, which resets the whole page.

SheetMonkey was used to send this value upon submission to a google sheets file, where I tracked and analysed the results.

5.6 Tools Developed

The tools developed were the following:

- **SVG Importer and Modeler**

  While incomplete as mentioned, it still gives most of the groundwork and is still usable in specific SVG's

- **3D Text Wrapper**

  This tool was extensively used in this project, as it is present on the content ice walls and on the book. Without it, all the text generated would be chaotic and hard to read. This tool enables any maker to effectively generate and organize text with features such as: wrapping, dynamic font resizing, text alignment and paragraph generation.

  If one wanted to, a full novel could be rendered using this tool.

- **Character State Machine**

  While state machines are not new in the slightest, there is no framework or add-on to generate these state machines for character animations.

- **JSON Dynamic Model Generation**

  This was applied on a very specific case, which is for the page generation and content ice wall generation, but both follow the same principle. Both can generate anything from any text in its respective format, making them viable reusable tools.

- **The Full Project**

  Finally, but definitely not least, this project as a whole. Just by switching the JSON content file and a couple of assets, the experience is completely different.

  It is dynamically generated and modular, so it can be the starting (and ending) point for any developer wanting to host their website in 3D.

Chapter 6

Evaluation

To evaluate the user's previous opinions, usage metrics and newly formed opinions, 2 questionnaires where created. Calls for experimenting the website and answer the respective forms were put through several ways: social media, online forums, online communities and the Faculty's mailing list. This was the best way to ensure the user testing base was large enough and varied enough to take any significant conclusions from the results.

The first one, named **Pre-Game Form**, aims to measure the users previous notions regarding 3D technology, to gauge their interests and digital spacial capabilities and their previous experience with 3D based websites.

The second and last form, named **Post-Game Form**, has the goal of re-access the same topics of interests and opinions the user presented in the first form, after the user explores the game to their desired extent, to analyse if there is a positive or negative trend. If positive, it would mean the experience had a positive impact in the topics it focused in. It also directly measures the users self perceived enjoyment, engagement and general perception of the game experience.

Finally, through the collection of usage period data, a graph and a few key statistics were drawn to compare regarding the average among websites.

However, before this final assessment, the project iterated through several stages with the use of direct user feedback.

6.1 Early Iterative Steps

With the objective of getting feedback as soon as possible, through the volunteering of fiends and strangers from online developer communities, I was able to show early prototypes for user testing.

There were no forms and the feedback was direct. While not the best approach from a research standpoint, it was the fastest way to reach the best final version of the project to then mass publicize and get a wide user testing base.

The feedback resulted in the following features: performance enhancing, controls smoothing, prompt clarification and audio focus.

Without this Agile based approach this project would not have reached the needed immersion levels.

6.2 Pre-Game Form

6.2.1 Questions

In this subsection I will describe the intent and objective of each question of the pre-game form.

Question 1 - *How often do you use 3D technology in your daily life?*

The first question starts by warming up the user towards the topic. It tries to measure how experienced the user is with 3D technology. This answer can skew the analysis as an overly 3D comfortable user testing pool would not be representative of the target audience and vice-versa.

Question 2 - *What 3D technologies have you used (experienced first or second hand) before?*

The second question aims to gauge the user testers regarding what specific areas of 3D they have tried or seen. This can help guide future work as to which themes and areas this type of website might be better used on.

Question 3 - *How comfortable are you navigating and interacting with 3D environments?*

The third question exists to make a comparison with the End Form, to see if the experience was accessible or if it exacerbated the users struggles regarding digital spacial awareness.

Question 4 - *What features or aspects of 3D technology do you enjoy the most?*

The forth question asks the user to list their pre-conceptions regarding what they usually enjoy the most regarding 3D technology.

Question 5 - *Have you ever used a 3D website before?*

The fifth question exists to see how experienced the user base is with this specific genre of website. It helps to better identify the testing users background.

Question 6 - *How likely are you to use websites with 3D technology in the future?*

The sixth question, and final question for those that answered negatively to the previous question, asks if they see themselves, in the future, using 3D websites after having an idea of what they entail from the questionnaire.

Question 7 - *What was your experience like using 3D websites?*

The seventh question is a simple measurement from experienced users to then compare to the projects End Form overall rating question.

Question 8 - *Would you like for 3D websites to be more common?*

The last question, tries to measure the users opinion regarding other 3D based websites. "Yes" means they like them in general, "Yes, if they have a lot of quality" means basic 3D experiences have probably had a negative effect, "No" means their experience was purely negative, and finally "No, but I still like them now and then" means they usually dislike them but one or another might prove otherwise.

6.2.2 Results

In this subsection I go over the results of the pre-game form, that had 111 participants.

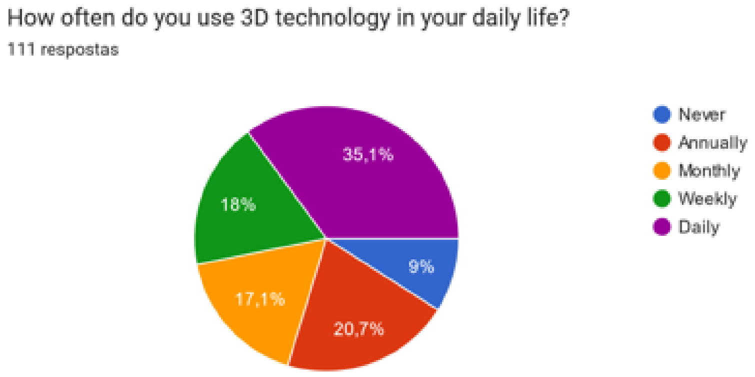


Figure 6.1: Start Form Question 1

Question 1 - The first results show that an overwhelming majority of users has regular experience with 3D technology, which is to be expected given its prevalence.

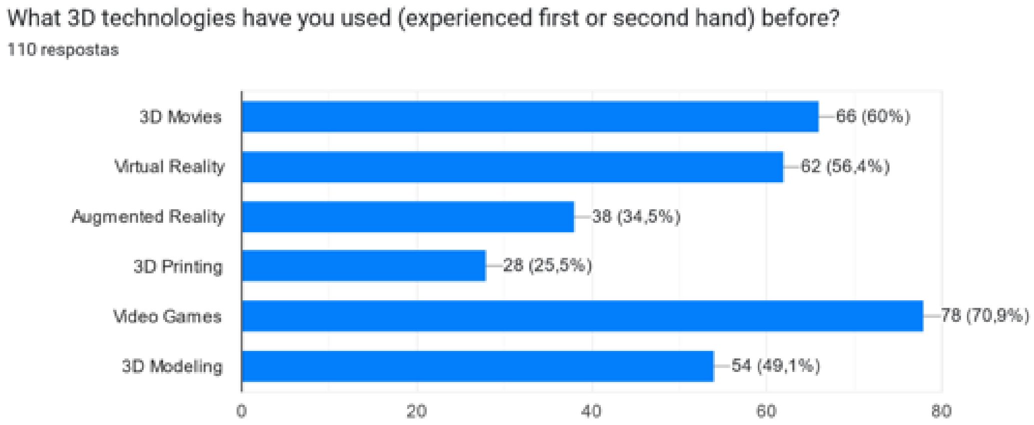


Figure 6.2: Start Form Question 2

Question 2 - Surprisingly, after Video Games and 3D Movies, Virtual reality was a close contender for the main technology. This shows that an investment in a VR compatible web experience could be fruitful.

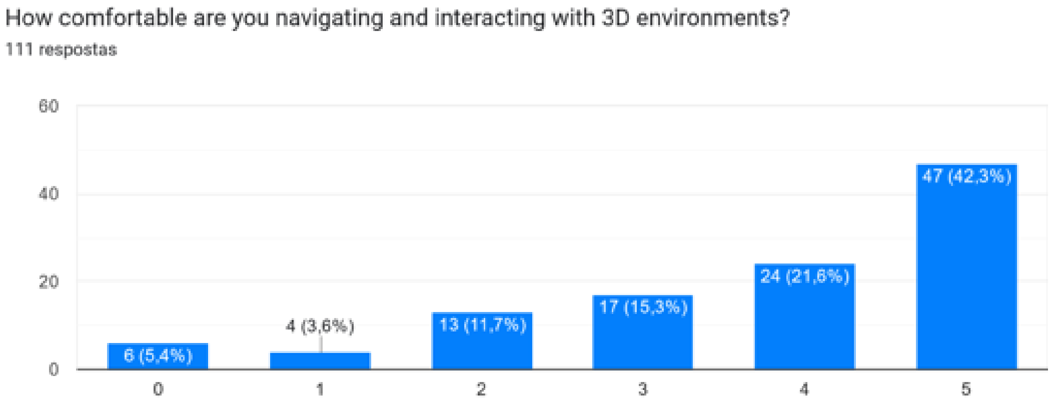


Figure 6.3: Start Form Question 3

Question 3 - Around 80% of users claim they are comfortable navigating and interacting with 3D environments (> 3). This is quite positive, as it suggests that, in recent years, this skill grew massively, meaning any 3D technology is now also more accessible.

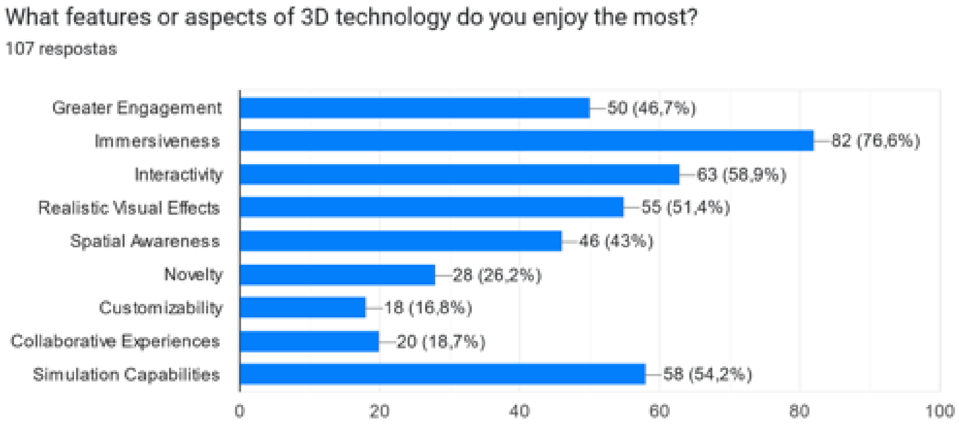


Figure 6.4: Start Form Question 4

Question 4 - The users show that the most important features, to them, on 3D tech are: immersiveness, interactivity, engagement and visual effects. The simulation capabilities was a surprise but, by linking it with the results from Question 2, it can be deduced it is from the overall Virtual Reality familiarity.

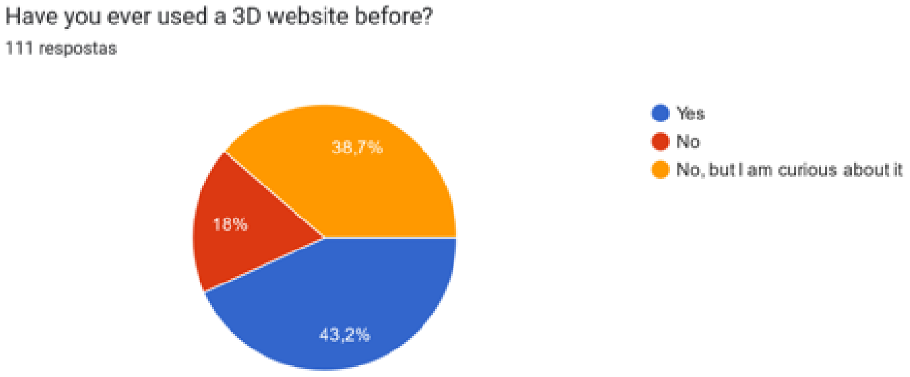


Figure 6.5: Start Form Question 5

Question 5 - While close to half, users that did experience a 3D website is a minority. This indicates that, even though 3D renders in 2D websites are getting more common over the past few years, 3D website technology is still under represented to this point. However, from those that did not experience one before, 2 out of 3 are also curious about what they could be like.

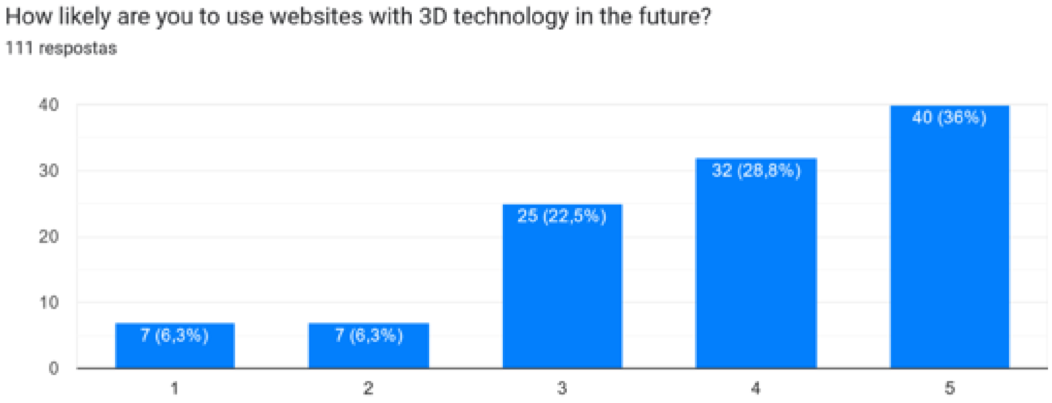


Figure 6.6: Start Form Question 6

Question 6 - Overall the consensus is it is likely they will indeed use 3D websites in the future, indicating the user base believes the future of website making will involve 3D components. Roughly a third of all answers express certainty that this will be the case.

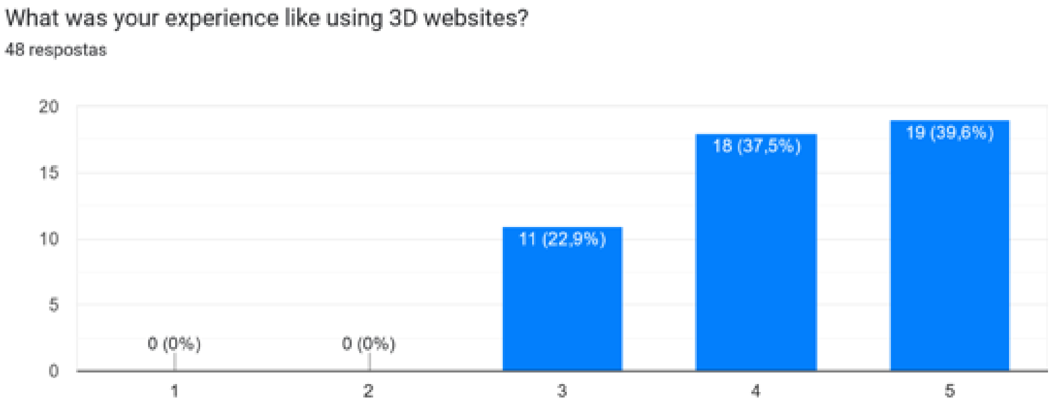


Figure 6.7: Start Form Question 7

Question 7 - This question was only presented to those that answered "Yes" to Question 5. There was not a single user that reported having negative experiences with 3D websites, which is telling of the potential the technology has.

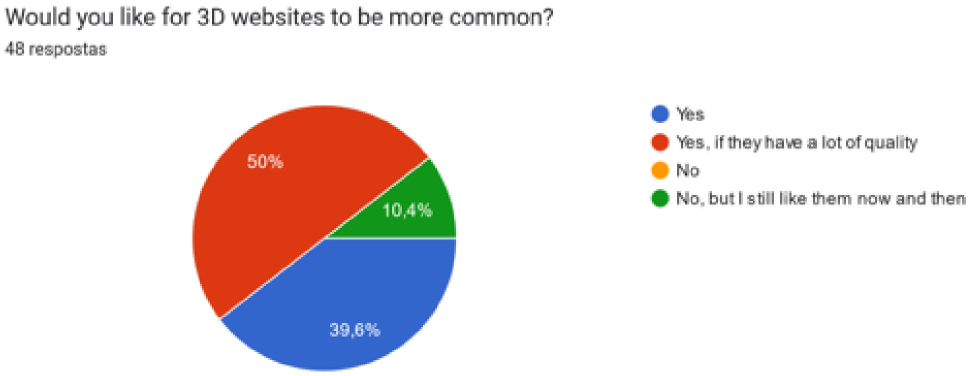


Figure 6.8: Start Form Question 8

Question 8 - This question was only presented to those that answered "Yes" to Question 5. 90% answered positively, with a small focus on quality, regarding the possible widespread of 3D websites. Only 10% answered negatively, but with a catered liking to some 3D websites. None wish for 3D web technology to not be more common.

6.3 Post-Game Form

6.3.1 Questions

In this subsection I will describe the intent and objective of each question of the post-game form.

Question 1 *Rate your experience with this website*

This question evaluates the overall experience of the user, to have a general evaluation of the experience.

Question 2 *Grade the following questions regarding your experience with the website*

This multi-part question gauges which aspects of the experience were positive, in a scale of 0 to 5, to have a ranking of areas of success and areas to work in the future.

Question 3 *How likely are you to visit this website again?*

Since it is hard to calculate an actual return rate, this question simulates it by asking what the likelihood of the user returning is.

Question 4 *What features or aspects of 3D technology used did you enjoy the most?*

This question asks, similarly to Question 4, the user to list their post-conceptions regarding what they liked the most regarding 3D technology, to compare.

Question 5 *Did this website leave a positive lasting impression or changed the way you think of 3D websites?*

This question aims to verify if, even if the user did not enjoy this experience, if it changed their prespective.

Question 6 *As a user, would you value websites that put work into personalized 3D experiences?*

To gauge if the experience changed the users opinions regarding 3D, it asks if they value 3D websites, 2D websites or a mix of both.

Question 7 *As a recruiter, would you value candidates that put work into a personalized 3D curriculum?*

The same as Question 6, but from a recruiter's perspective, as the target audience for a curriculum website is a recruiter.

Question 8 *How likely are you to use websites with 3D technology in the future?*

To get an idea of the users perspective on the future, this question asks the likelihood of encountering fully or partially 3D websites.

Question 9 *Would you like for 3D websites to be more common?*

To round it all up, this question evaluates if the user wants this technology to be more widespread.

6.3.2 Results

In this subsection I go over the results of the post-game form, that had 50 participants (vastly less than the pre-game questionnaire).

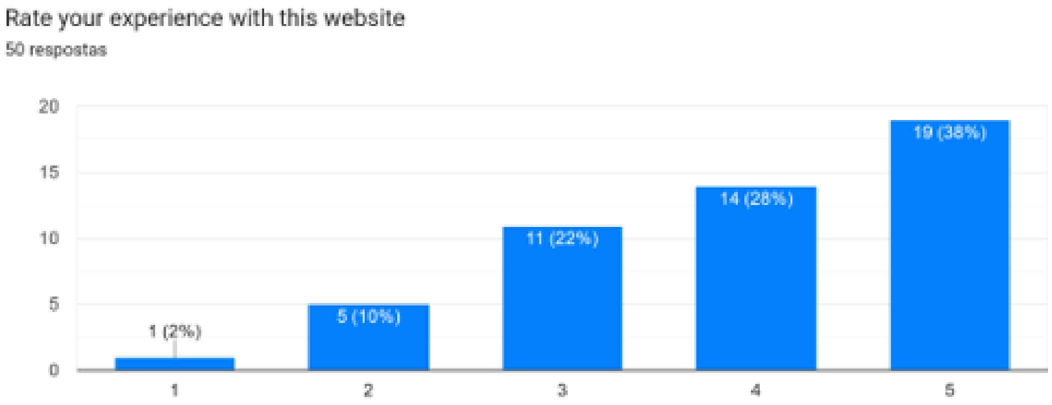


Figure 6.9: End Form Result 1

Question 1 - The average rating was of 4, so the result was very positive, indicating that the project and proposal were successful.

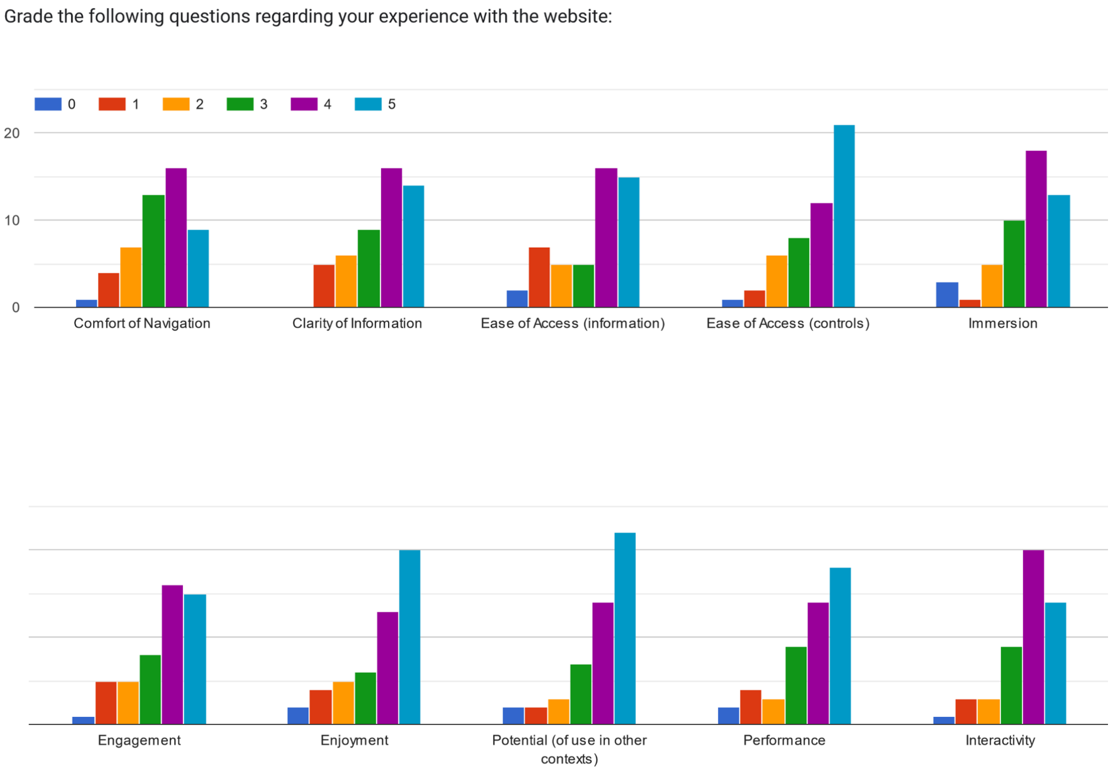


Figure 6.10: End Form Result 2

Question 2 - This question has a couple of different areas to tackle on the results. Firstly, ease of access (controls), potential, enjoyment and performance stand out as having (or close to having) a majority on the 5/5 score. Secondly, comfort of navigation, clarity of information and immersion are the apparently worst categories, but even so they are, no average, quite positive. Lastly, the other categories have a generally good average (between 4 and 5). All of this shows that all categories managed to have a good reception, proving the success of the proposal.

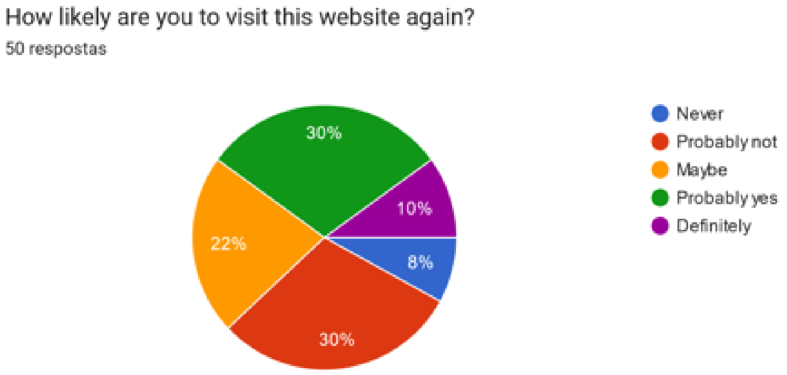


Figure 6.11: End Form Result 3

Question 3 - The simulated return rate shows that around 40% of users would return to this website (10% with certainty). On the other hard, around 40% would also not return to the website (8% with certainty). The remaining 22% are indecisive. This can be approximated to a return rate of around 50%. While this might seem low, it is actually much above the average return rate for CV websites, as they tend to be a website each user only visits once. So it was an overwhelming success in the return rate increase.

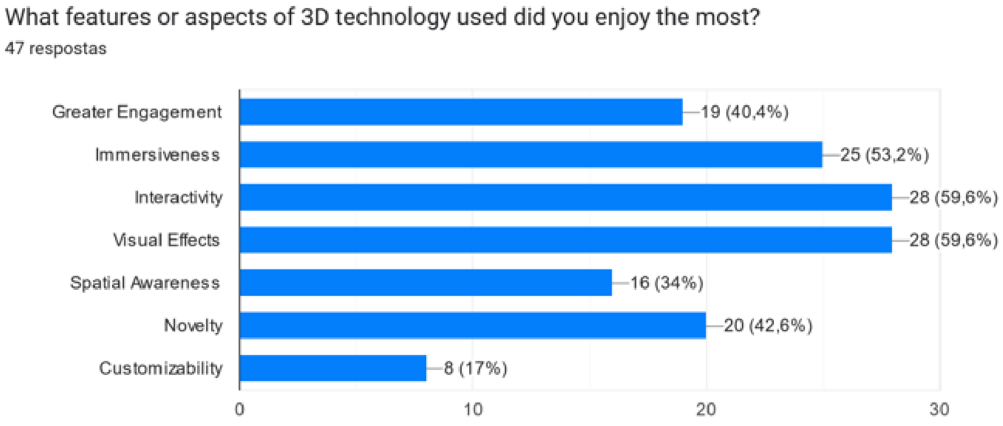


Figure 6.12: End Form Result 4

Question 4 - The clear winners were the immersiveness, visual effects and interactivity of the website. Even so, over a third chose all but one option, showing the impact it had on users.

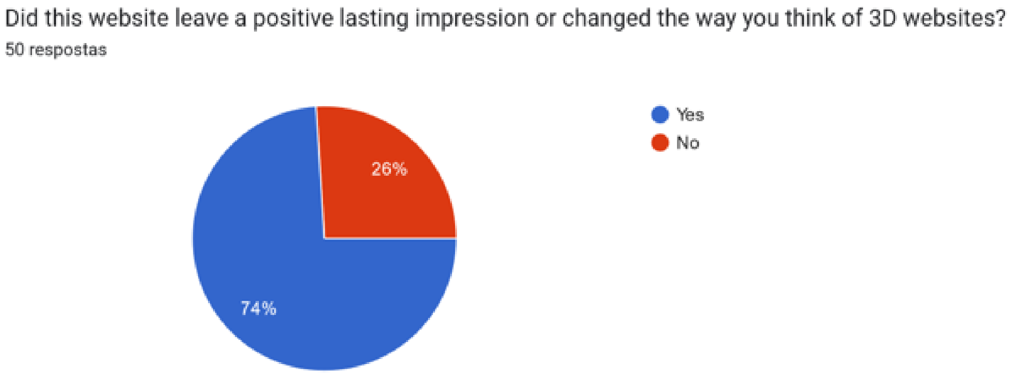


Figure 6.13: End Form Result 5

Question 5 - With a 74% positive response, the project, once again, proved successful in it’s objectives.

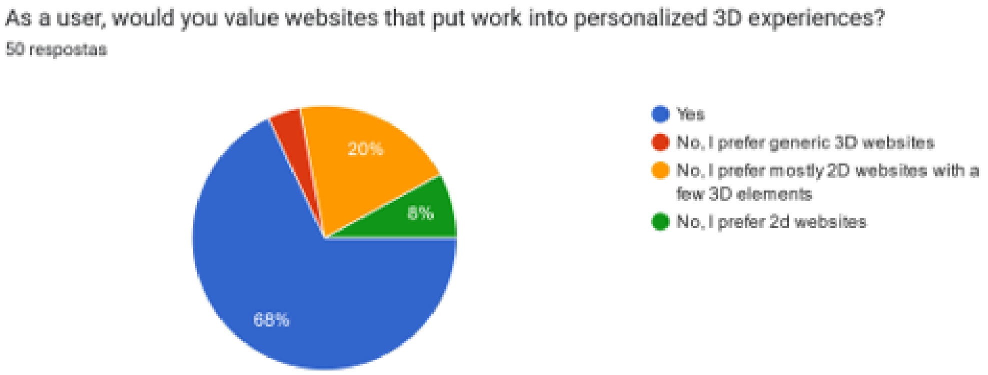


Figure 6.14: End Form Result 6

Question 6 - 68% of users would value more websites of the sort, meaning this experience was extremely positive and enjoyable. Of the rest that has other preferences, most still value 3D elements, as only 8% said they only want fully 2D websites.

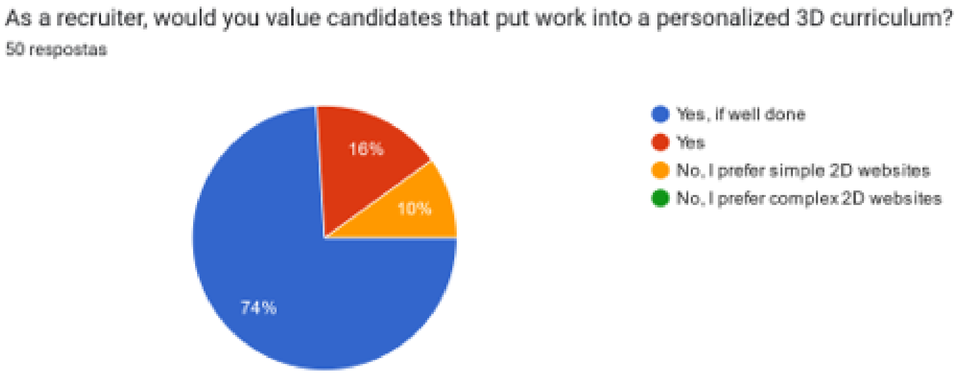


Figure 6.15: End Form Result 7

Question 7 - An overwhelming 90% said that, as recruiters, they would value 3D curriculums, further proving the project's success.

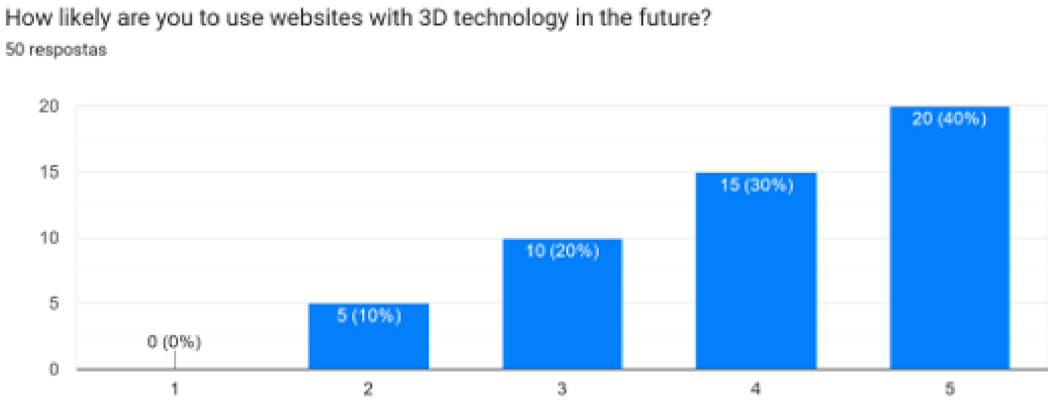


Figure 6.16: End Form Result 8

Question 8 - The vast majority (90%) voted 3 or above, meaning they see this technology as the future.

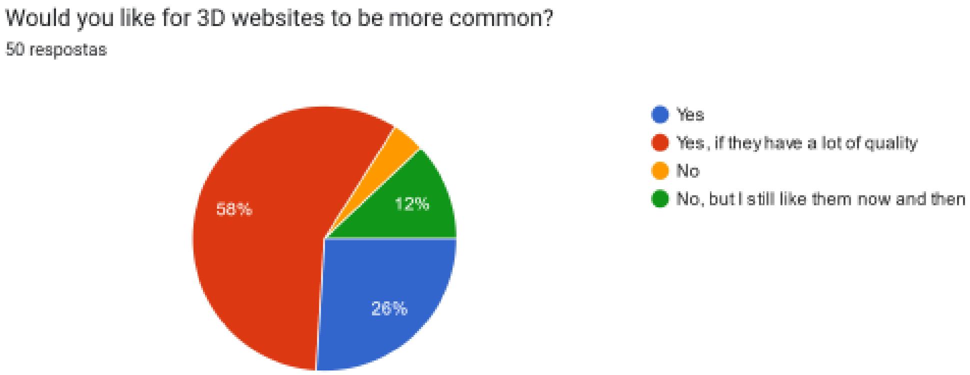


Figure 6.17: End Form Result 9

Question 9 - Lastly, akin to Question 6, 85% would want 3D websites to be more common. 12% would still like them once in a while.

6.4 Time Spent Metric Results

To measure time spent, a hidden HTML form tracked the users time in the website from the end of the Start Form, to the start of the End Form. This was done as the time spent answering the forms should be disregarded as they are not part of the project experience.

These values were submitted and saved on a Google Sheets page and then statistically analysed to compare to the standard metrics of websites.

A major setback is the inability to save the time spent when a user leaves the website without answering the End Form. This was a problem as only 50% of test users that answered the First Form actually submitted the End Form (52 people in total).

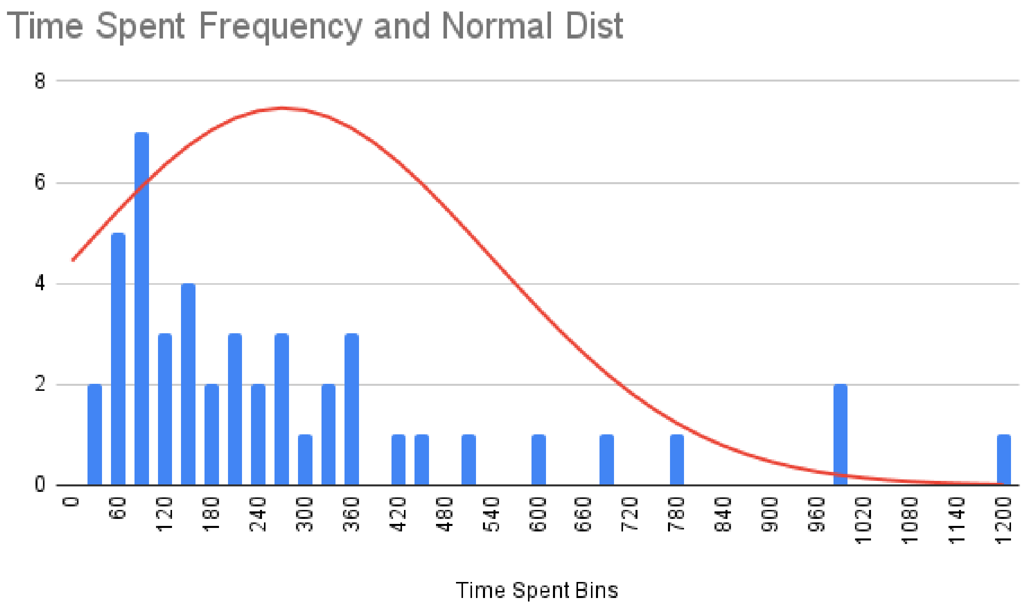


Figure 6.18: Time Spent Graph

The data was firstly aggregated in buckets of 30 (X axis), as all values were unique due to the millisecond accuracy (as seen on the Time Spent Graph). Then the frequency of each bucket was charted (Y axis).

The **Average Time Spent** equated to **272 seconds**, with a **194 seconds Median** and **267 seconds Standard Deviation**.

The large standard deviation can be attributed to the large positively **skewed normal distribution** curve, visualized on the Time Spent Graph. This is expected as the values can be extremely large, while being more common for them to be closer to a low value.

6.5 Analysis and Comparison

To streamline this section, it was divided into 3 separate areas of evaluation:

6.5.1 User Satisfaction

Questions 1, 5 and the comments optional field show an overwhelming positive feedback regarding enjoyment and satisfaction. While there are improvements to be made, such as performance, ease of access to information (regarding time) and some information clarity issues, these only account for less than 25% of feedback and most was given while the experience overall was still positive.

It is also important to note that most (over 80%) negative feedback came from users that experienced poor performance, so the data relative to user satisfaction is actually better than shown.

By comparing most questions from the Start Form and End Form, other than 5 which shows an overwhelming positive change in user opinion regarding 3D websites, we can say that the experience lived up to users expectations, as no significant changes occured from the Start Form to the End Form, while having a very positive feedback (see Start Questions and 8, and End Questions 4 and 9).

Thus, regarding the increase of user satisfaction, the project was a huge success.

6.5.2 User Engagement

According to a Forbes publication that aggregated many website statistics, *"Website users spend an average of 5.59 seconds looking at written content on a site"*.

Mainly it states that *"The average time spent on a page is 54 seconds"*, which means the project managed to increase user engagement time by over **5 times** the average. This is an incredible result that shows that, even though not perfect, the website was incredibly successful in achieving this goal.

According to the users reviews on questions 2 and 4, this statistical result was due to the ease of use and access, high immersion (due to the aesthetics, sound/visual feedback and smooth navigation), lots of interactivity and an overall good performance.

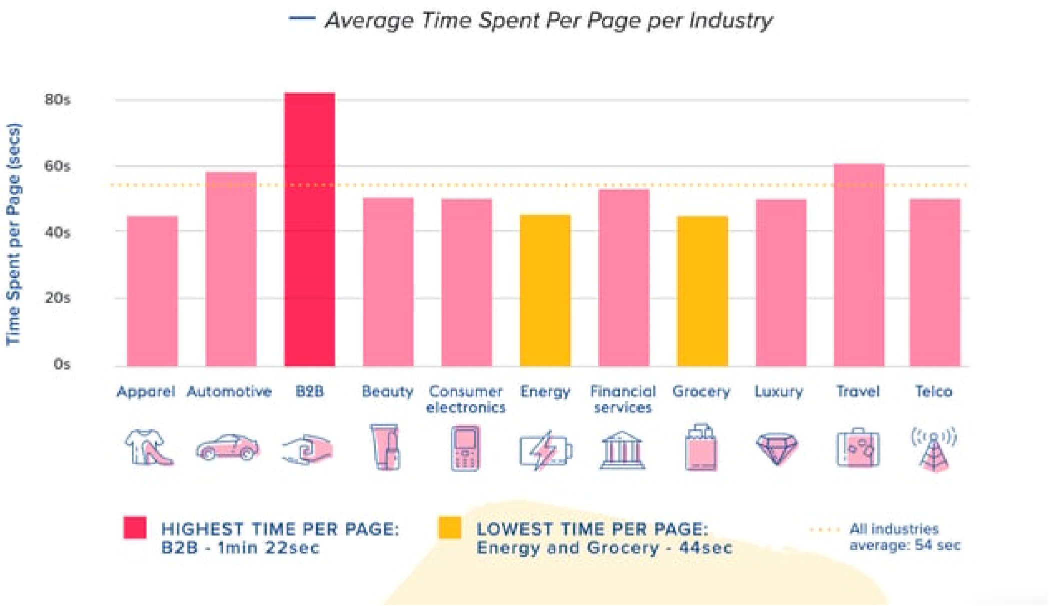


Figure 6.19: Average Time Spent on Websites per Industry [14]

HubSpot, another source, gives a more detailed average metric, aggregated by industry type. This graph goes inline with Forbe's statistic and showing that, even on the industries with higher user engagement, this website still stands easily above.

Forbes also mentioned *"Including videos on a website can increase time spent on page by 88%"*[6], which equates to an average of 101 seconds spent. While closer to our results, this proves that fully immersive 3D content engages the user for longer than videos.

6.5.3 User Retention

"88% of online users won't return to a site after a bad experience"[6], quoted by Forbes and sourced by Sweor, shows the importance of user satisfaction towards user retention. The more a user enjoys the website, the more likely it is for them to return in the future.

While this is attributed in a large part to the website content, most users value first impressions more: *"61% say that if they don't find what they're looking for within about five seconds, they'll go to another site"*[6]. Users that leave in these early parts are not likely to return.

According to the results on question 3, roughly 22% reported uncertainty if they would visit this website again. However, from the remaining, half would indeed want to return to this website. This is an outstanding result as CV websites are not prone to being revisited (recruiters tend to see them once, get the information needed, and never come back).

Questions 1 and 5 provide groundwork to justify the willingness to return, as 74% reported a positive lasting impression from using this website and 50% had great joy in experiencing this website. A noteworthy fact is that most (over 80%) negative feedback came from users that experienced poor performance, so the data relative to user satisfaction is actually better than shown.

6.6 Conclusion

Both the user questionnaires and time metrics overwhelmingly support several take-aways:

- Users want 3D technology to be more wide spread.
- Users enjoy catered and personalized experiences, mainly if they are 3D.
- Users spend more time on 3D gamified websites than average.
- Users enjoy a break from traditional web design, valuing the 3D display of 2D information.
- Users are more likely to return to this type of website, due to a combination of higher engagement and satisfaction.

With this, it is concluded that the project was a massive success in both proving the potential of displaying 2D information in a gamified 3D environment and providing groundwork and tools towards other developers what want to delve in this specific area.

Chapter 7

Conclusions and Future Work

7.1 Project Conclusions

What started as a self proposed thesis idea, ended up as a 4 month programming project that challenged every part of me. It was chaotic at times, having to scrap a whole month's work midway.

Taking into account all the goals set from the previous state of the art research and planning, all of them were reached and, from the general feedback, they all served its designated purpose. Only some extras I thought of along the way managed to not get in the final version. Going over each objective:

- The 3D web game managed to be dynamic, by converting JSON information in 3D assets, allowing for the extremely easy and fast change of information.
- This dynamism also allows any developer or user to transform their 2D information into an 3D interactive and immersive experience.
- Through the modular design, all classes developed are available to be used independently by the open-source community, saving time and effort to any developer delving into this matter.
- Similarly, with the tools developed, such as the SVG Modeler, JSON Parser and Generator and 3D Text Wrapper, any developer using Three.js can now achieve similar results with greater ease and speed.
- With the documented project structure and full open-source availability, it provides a guideline on how to create a similar project with close results which, as will be mentioned next, were very positive.
- Also mentioned next, the usage of user experience, engagement and gamification principles was proven very effective in making an enjoyable and attention grabbing website in 3D.

- Finally, by making this concept work in a website type as critical and demanding as Curriculum websites, it further strengthens the results and usability of the tools developed.

While it was incredibly time consuming and hard, I believe the results speak for itself. With over 2500 lines of code, its a fully dynamic, easy to maintain, modular and expandable, immersive website that impressed many people already.

I can say I am incredibly proud of this website and I will keep maintaining and updating it as my online portfolio.

7.2 Research Conclusions

Overall the research had very positive results. While there was a minority that graded the experience negatively (< 3), over 90% of them indicated heavy performance issues. This is actually a good thing as it indicates that the few with very outdated hardware had a bad experience but it was all independent of the actual game and its design. Also, as this thesis project is meant to serve as groundwork for the future, it starts with the assumption of what the web could be in 5 years time, when hardware already got drastically better, further diminishing the importance of those specific answers.

- The website was able to incorporate all possible principles from user experience, such as consistency, visual hierarchy and memorability, allowing for a streamlined and accessible experience.

- By following a close design to a third person exploring game, all but the social interaction principle of gamification were implemented, pushing users to explore and spend more time on the website.

- With the results from the forms, the vast majority (96% of users, if those with performance issues are discarded) were very enthralled by the project, reporting increases in interest, return rates and overall enjoyment.

- Gamification and user experience, as bases for 3D website design, proved very useful, with quantifiable benefits as proven by user feedback and time spent metrics. Of all aspects, gamification related immersion, effects and interactivity were the most prominent as user reports suggest. Similarly, user experience related consistency, aesthetics, usability and visual hierarchy were key to assuring the success of the web application.

- The time spent metrics ended up far exceeding what was pretended, reaching over 5x the average, in a website type that tends to be incredibly monotonous.

- By casting a wide net through the spread of this experiment, the user base ended up very heterogeneous, giving the results more veracity and proving their usability.

Most of the state of the art research that was supposed to be proved in this context through the forms and metrics was successful, as the vast majority felt immersed, engaged and enjoyed the novelty of navigating a 3D environment built around 2D content.

7.3 Future Work

Regarding the project in the presented state, whats most important is to finish the **SVG Importer and Modeler** and the `Wall` Character State.

Another must have is the adaptation to mobile. Mobile represents the vast majority of internet traffic and, as such, a website like this must have a version for such. While the website loads on mobile and can run over 30fps, there are still no controls.

Thirdly, publishing the tools as a JavaScript and/or NPM module to allow the easy access from the open-source community, as well as opening some GitHub repositories for those who may want to contribute to their further evolution.

Taking into account the final research results, more can be done to further improve the users experience, such as: shortening the distance from the start area to the main area (possibly even eliminating the gap), improve the performance even further (for example, by adding occlusion culling and level of detail), making the directions and prompts even clearer as an option, adding a menu with game and visual settings, adding a quick character customization in the beginning and making the game VR compatible.

Finally, while the game and code structure is already good, some refactoring could be done to make it easily readable.

References

[1] 3d web - explore - google trends. https://trends.google.com/trends/explore?date=all&q=3D%20web.

[2] adobe flash - explore - google trends. https://trends.google.com/trends/explore?date=all&q=adobe%20flash.

[3] gamification - explore - google trends. https://trends.google.com/trends/explore?date=all&q=gamification.

[4] How web development has changed over the last 10 years | appmaster. https://appmaster.io/blog/how-web-development-has-changed-the-last-10-years.

[5] three.js - explore - google trends. https://trends.google.com/trends/explore?date=all&q=three.js.

[6] Top website statistics for 2023 – forbes advisor. https://www.forbes.com/advisor/business/software/website-statistics/.

[7] user engagement - explore - google trends. https://trends.google.com/trends/explore?date=all&q=user%20engagement.

[8] user experience - explore - google trends. https://trends.google.com/trends/explore?date=all&q=user%20experience.

[9] user experience - explore - google trends. https://trends.google.com/trends/explore?date=all&q=user%20experience.

[10] Ux/ui - explore - google trends. https://trends.google.com/trends/explore?date=all&q=UX%2FUI.

[11] vr - explore - google trends. https://trends.google.com/trends/explore?date=all&q=vr.

[12] Web design is now completely boring. https://thenextweb.com/news/zzzzzz.

[13] Web development evolution from the 2000s' to 2020 | jellyfish.tech. https://jellyfish.tech/blog/web-development-evolution-from-2000s-to-2020/.

[14] What is the average time spent on a website? [+ how to improve it]. https://blog.hubspot.com/marketing/chartbeat-website-engagement-data-nj.

[15] Naim Asaj, Bastian Könings, Mark Poguntke, Florian Schaub, Björn Wiedersheim, and Michael Weber. Gamification: State of the art definition and utilization. *d-nb.info*, 2012.

[16] José Azevedo, Patrícia Padrão, Maria J Gregório, Carla Almeida, Nuno Moutinho, Nanna Lien, and Renata Barros. A web-based gamification program to improve nutrition literacy in families of 3-to 5-year-old children: the nutriscience project. *Journal of nutrition education and behavior*, 51(3):326–334, 2019.

[17] Albert N. Badre. Shaping web usability. *Ubiquity*, 2002:1, 2 2002.

[18] Rakesh Baruah and Rakesh Baruah. Diving into three. js. *AR and VR Using the WebXR API: Learn to Create Immersive Content with WebGL, Three. js, and A-Frame*, pages 123–169, 2021.

[19] David Benyon. *Designing user experience*. Pearson UK, 2019.

[20] Judith Borghouts, Elizabeth Eikey, Gloria Mark, Cinthia De Leon, S. M. Schueller, Margaret Schneider, Nicole Stadnick, Kai Zheng, Dana Mukamel, and Dara H. Sorkin. Barriers to and facilitators of user engagement with digital mental health interventions: Systematic review. *J Med Internet Res 2021;23(3):e24387 https://www.jmir.org/2021/3/e24387*, 23:e24387, 3 2021.

[21] I Caponetto, J Earp, M Ott European Conference on Games, and undefined 2014. Gamification and education: A literature review. *books.google.com*.

[22] Omar López Chávez, Luis-Felipe Rodríguez, and J Octavio Gutierrez-Garcia. A comparative case study of 2d, 3d and immersive-virtual-reality applications for healthcare education. *International journal of medical informatics*, 141:104226, 2020.

[23] Chih-Hung Chung and You Yin Lin. Online 3d gamification for teaching a human resource development course. *Journal of Computer Assisted Learning*, 38(3):692–706, 2022.

[24] Piotr Adam Cybulski. *A comparison of Unity and Three. js in terms of visualizing data from ALICE detector in a web browser*. PhD thesis, Instytut Informatyki, 2021.

[25] Farrah Diana, Saiful Bahry, Maslin Masrom, and Mohamad Noorman Masrek. Website credibility and user engagement: A theoretical integration. *Proceedings - 2016 4th International Conference on User Science and Engineering, i-USEr 2016*, pages 216–221, 2 2017.

[26] Björn Fischer, Alexander Peine, Britt Östlund, and Patricia C. Heyn. The importance of user involvement: A systematic review of involving older users in technology design. *The Gerontologist*, 60:e513–e523, 9 2020.

[27] J Hamari, J Koivisto, H Sarsa 2014 47th Hawaii international, and undefined 2014. Does gamification work?–a literature review of empirical studies on gamification. *ieeexplore.ieee.org*, 2014.

[28] Juho Hamari. Gamification. *The Blackwell Encyclopedia of Sociology*, pages 1–3, 11 2019.

[29] HCL Hsieh, HH Yang TheoreTical issues in ergonomics science, and undefined 2020. Incorporating gamification into website design to facilitate effective communication. *Taylor Francis*, 21:89–111, 1 2019.

[30] Shang Hwa Hsu, Jen Wei Chang, and Chun Chia Lee. Designing attractive gamification features for collaborative storytelling websites. *Cyberpsychology, Behavior, and Social Networking*, 16:428–435, 6 2013.

[31] Benjamin Kenwright. Visualization with three. js. In *12th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia 2019*, 2019.

[32] Firas Layth Khaleel, Tengku Siti Meriam Tengku Wook, and Noraidah Sahari Ashaari. Quantifying user experience in using learning gamification website. *jatit.org*, 15, 2018.

[33] FL Khaleel, NS Ashaari, TSMT Wook Jurnal Teknologi, and undefined 2019. An empirical study on gamification for learning programming language website. *researchgate.net*, 2019.

[34] G Kiryakova, N Angelova, L Yordanova Proceedings of 9th, and undefined 2014. Gamification in education. *academia.edu*.

[35] Ana Carolina Tomé Klock, Isabela Gasparini, Marcelo Soares Pimenta, and Juho Hamari. Tailored gamification: A review of literature. *International Journal of Human-Computer Studies*, 144:102495, 2020.

[36] Jeanine Krath, Linda Schürmann, and Harald FO Von Korflesch. Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. *Computers in Human Behavior*, 125:106963, 2021.

[37] Ahyun Lee and Insung Jang. Implementation of an open platform for 3d spatial information based on webgl. *ETRI Journal*, 41(3):277–288, 2019.

[38] Zhen Liu, Xingyu Gu, Qiao Dong, Shanshan Tu, and Shuwei Li. 3d visualization of airport pavement quality based on bim and webgl integration. *Journal of Transportation Engineering, Part B: Pavements*, 147(3):04021024, 2021.

[39] Jesús López-Belmonte, Mª Elena Parra-González, Adrián Segura-Robles, and Santiago Pozo-Sánchez. Scientific mapping of gamification in web of science. *mdpi.com*.

[40] Fiona Fui-Hoon Nah, Qing Zeng, Venkata Rajasekhar Telaprolu, Abhishek Padmanabhuni Ayyappa, and Brenda Eschenbrenner. Gamification of education: a review of literature. *Springer*, 8527:401–409, 2014.

[41] Mishal Hasnain Naqvi, Sun Guoyan, and Muhammad Hasnain Abbas Naqvi. Measuring the influence of web features in the online gamification environment: a multimediation approach. *Wireless Communications and Mobile Computing*, 2021:1–17, 2021.

[42] Winda Suci Lestari Nasution and Patriot Nusa. Ui/ux design web-based learning application using design thinking method. *ARRUS Journal of Engineering and Technology*, 1(1):18–27, 2021.

[43] Sebastian Oberdörfer, David Heidrich, and Marc Erich Latoschik. Usability of gamified knowledge learning in vr and desktop-3d. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[44] Heather L. O'Brien and Elaine G. Toms. What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology*, 59:938–955, 4 2008.

[45] Heather L. O'Brien and Elaine G. Toms. The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61:50–69, 1 2010.

[46] G Aydin International Journal of Online Marketing (IJOM) and undefined 2015. Adoption of gamified systems: a study on a social media gamification website. *igi-global.com*, 2015.

[47] Jeeyun Oh and S. Shyam Sundar. User engagement with interactive media: A communication perspective. *Why Engagement Matters: Cross-Disciplinary Perspectives of User Engagement in Digital Media*, pages 177–198, 1 2016.

[48] Michael Sailer and Lisa Homner. The gamification of learning: A meta-analysis. *Educational Psychology Review*, 32(1):77–112, 2020.

[49] Awaz Naaman Saleem, Narmin Mohammed Noori, and Fezile Ozdamli. Gamification applications in e-learning: A literature review. *Technology, Knowledge and Learning*, 27(1):139–159, 2022.

[50] Dick Schoech, Javier F. Boyas, Beverly M. Black, and Nada Elias-Lambert. Gamification for behavior change: Lessons from developing a social, multiuser, web-tablet based prevention game for youths. *Journal of Technology in Human Services*, 31:197–217, 2013.

[51] K Seaborn, DI Fels International Journal of human-computer studies, and undefined 2015. Gamification in theory and action: A survey. *Elsevier*.

[52] Michael Shane, Lukman Wisnudrajat, Gunawan Wang, and Emil Robert Kaburuan. Implementing gamification on a web-based recruitment system using marczewski's gamification framework: An overview. *International Journal*, 9(3):2667–2672, 2020.

[53] Y Xu CSDL Technical and undefined 2012. Literature review on web application gamification and analytics. *Citeseer*, 2012.

[54] Graham Wakefield. Webxr template for three. js and node. js. 2022.

[55] Peiling Wang, William B. Hawk, and Carol Tenopir. Users' interaction with world wide web resources: an exploratory study using a holistic approach. *Information Processing Management*, 36:229–251, 3 2000.

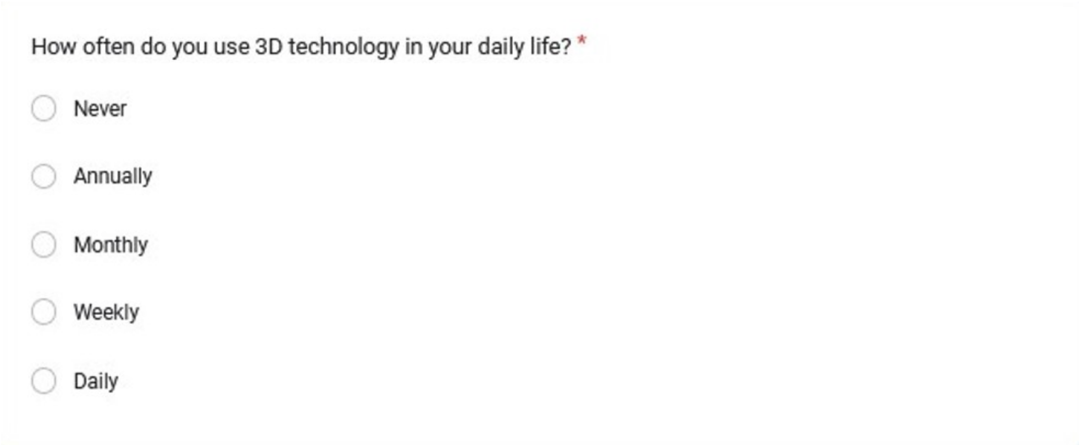
[56] Xiaoping Zhou, Jia Wang, Ming Guo, and Zhe Gao. Cross-platform online visualization system for open bim based on webgl. *Multimedia Tools and Applications*, 78:28575–28590, 2019.

[57] G Zichermann and C Cunningham. *Gamification by design: Implementing game mechanics in web and mobile apps*. 2011.

Appendix A

Pre-Game Questionnaire

A.0.1 Questions



How often do you use 3D technology in your daily life? *

○ Never

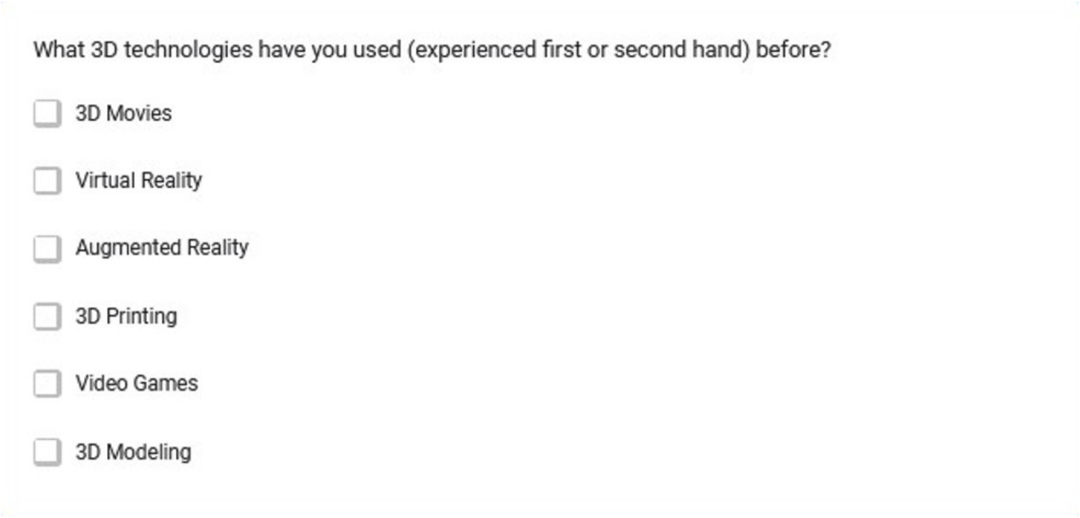
○ Annually

○ Monthly

○ Weekly

○ Daily

Figure A.1: Start Form Question 1



What 3D technologies have you used (experienced first or second hand) before?

□ 3D Movies

□ Virtual Reality

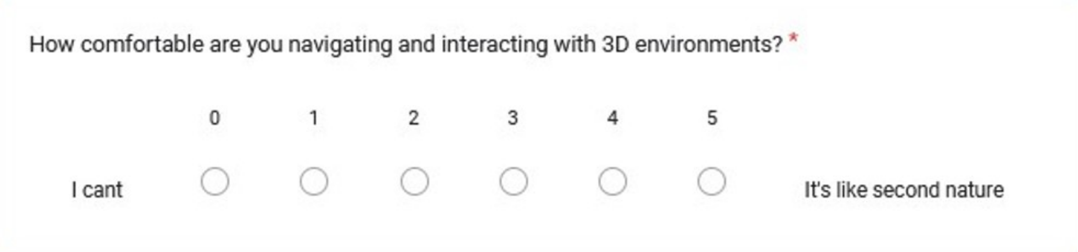
□ Augmented Reality

□ 3D Printing

□ Video Games

□ 3D Modeling

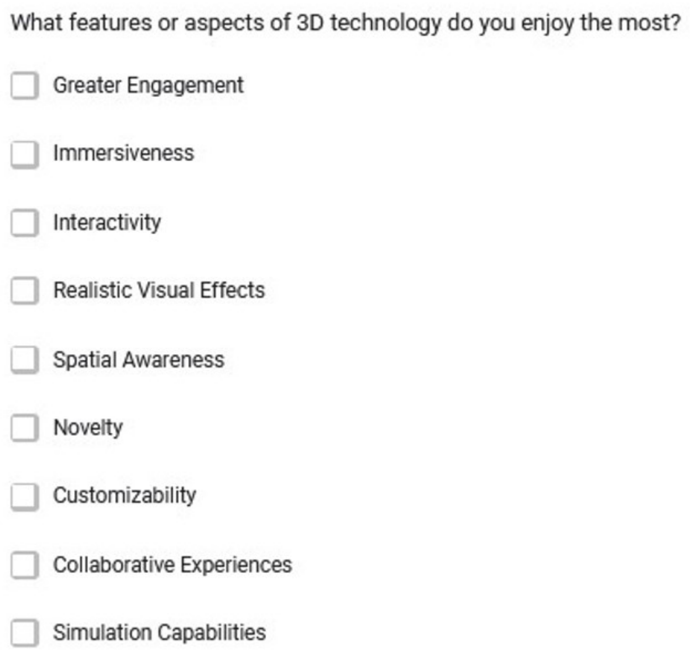
Figure A.2: Start Form Question 2



How comfortable are you navigating and interacting with 3D environments? *

	0	1	2	3	4	5	
I cant	○	○	○	○	○	○	It's like second nature

Figure A.3: Start Form Question 3



What features or aspects of 3D technology do you enjoy the most?

□ Greater Engagement

□ Immersiveness

□ Interactivity

□ Realistic Visual Effects

□ Spatial Awareness

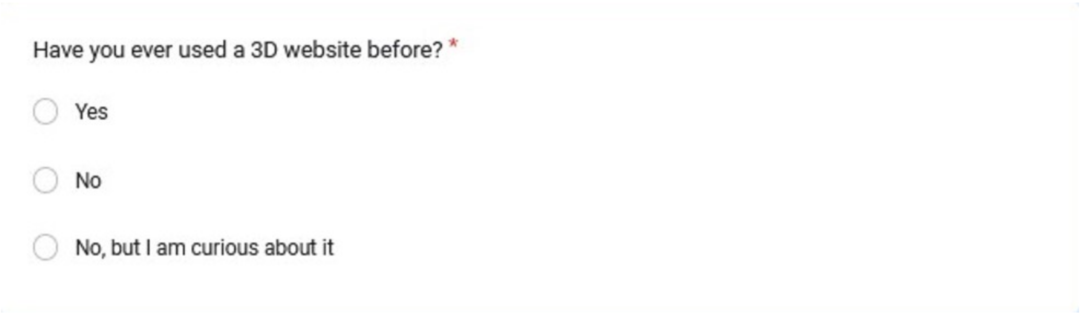
□ Novelty

□ Customizability

□ Collaborative Experiences

□ Simulation Capabilities

Figure A.4: Start Form Question 4



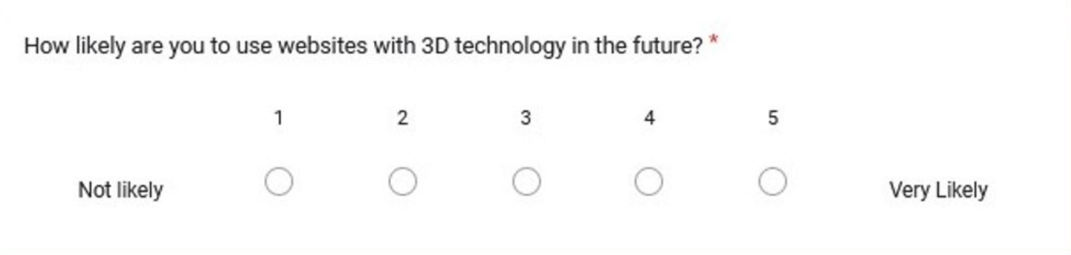
Have you ever used a 3D website before? *

○ Yes

○ No

○ No, but I am curious about it

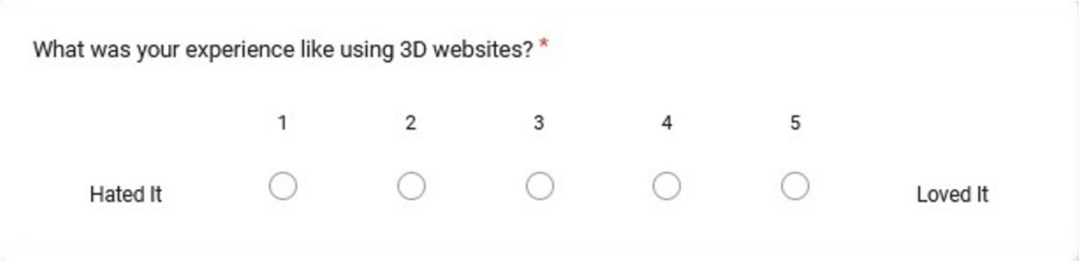
Figure A.5: Start Form Question 5



How likely are you to use websites with 3D technology in the future? *

	1	2	3	4	5	
Not likely	○	○	○	○	○	Very Likely

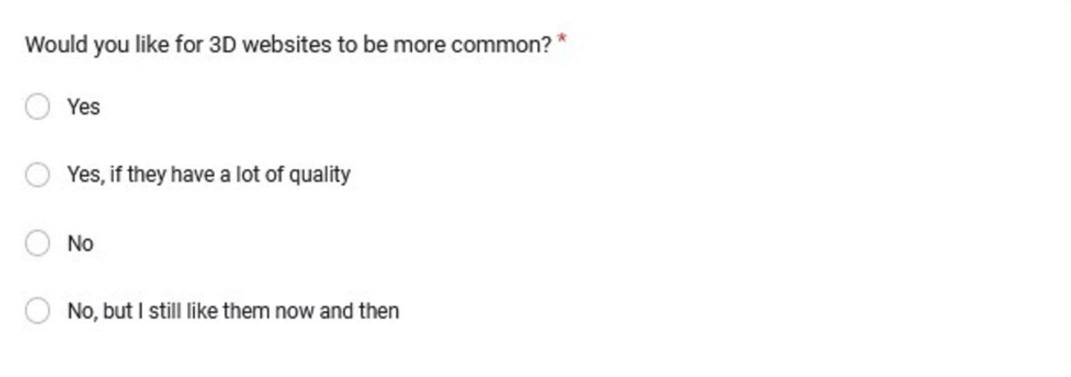
Figure A.6: Start Form Question 6



What was your experience like using 3D websites? *

	1	2	3	4	5	
Hated It	○	○	○	○	○	Loved It

Figure A.7: Start Form Question 7



Would you like for 3D websites to be more common? *

○ Yes

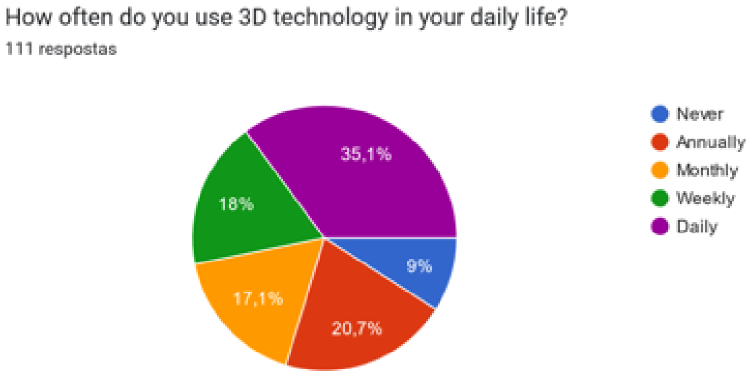
○ Yes, if they have a lot of quality

○ No

○ No, but I still like them now and then

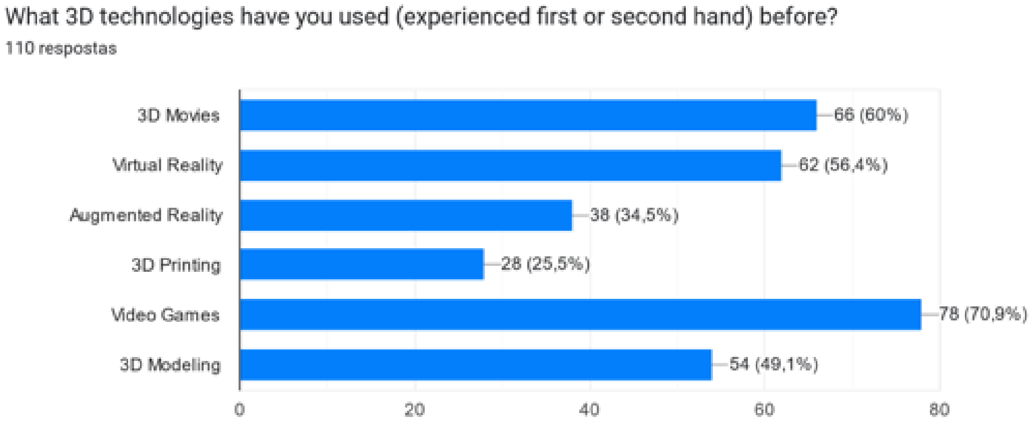
Figure A.8: Start Form Question 8

A.0.2 Results


How often do you use 3D technology in your daily life?

111 respostas

Figure A.9: Start Form Question 1


What 3D technologies have you used (experienced first or second hand) before?

110 respostas

Figure A.10: Start Form Question 2

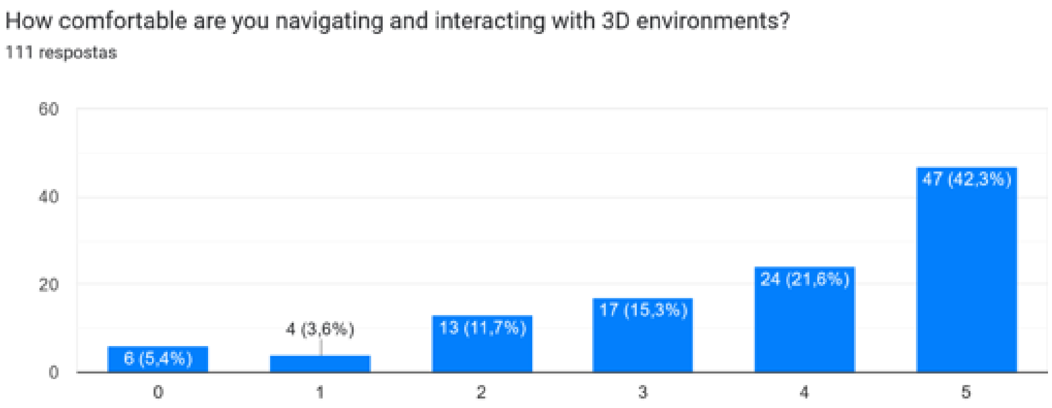


Figure A.11: Start Form Question 3

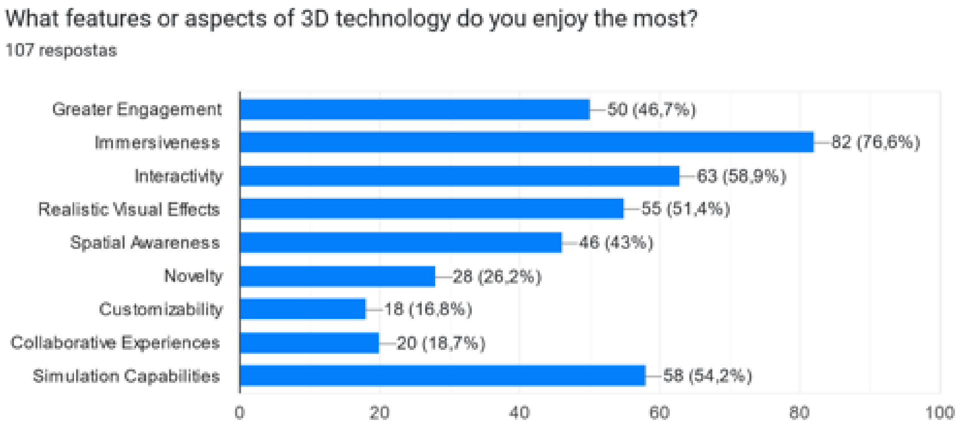


Figure A.12: Start Form Question 4

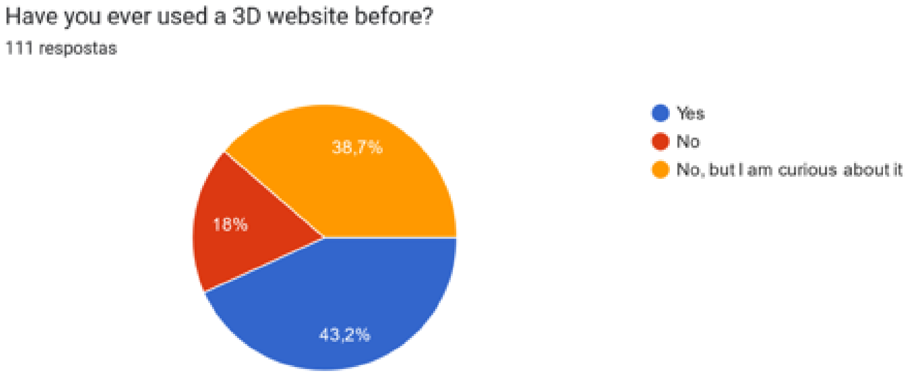


Figure A.13: Start Form Question 5

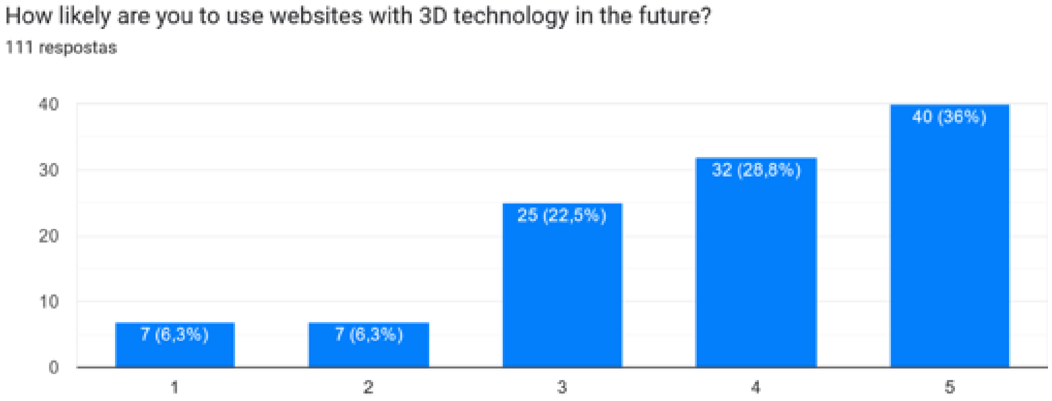


Figure A.14: Start Form Question 6

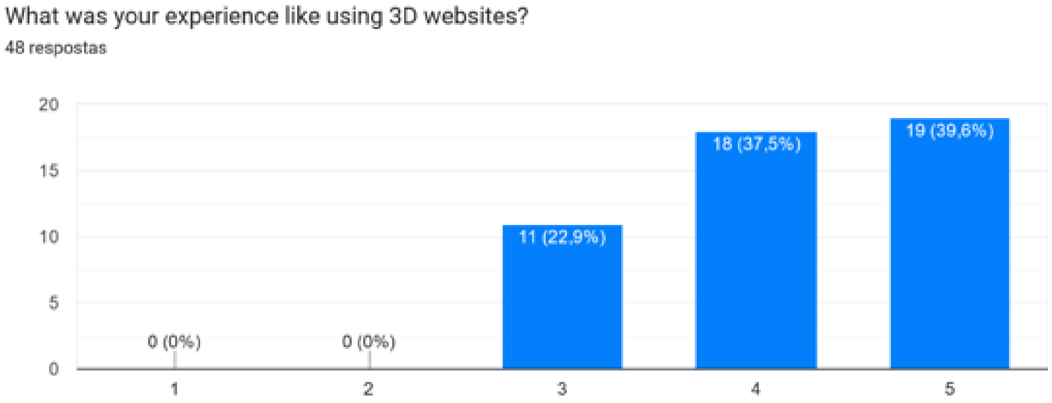


Figure A.15: Start Form Question 7

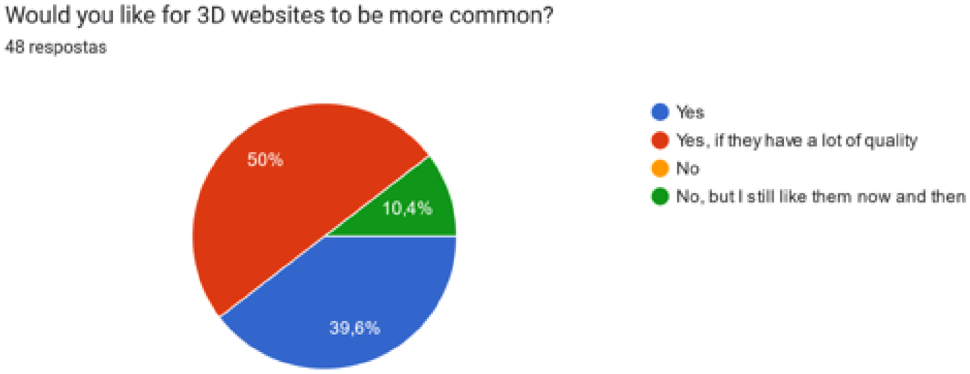
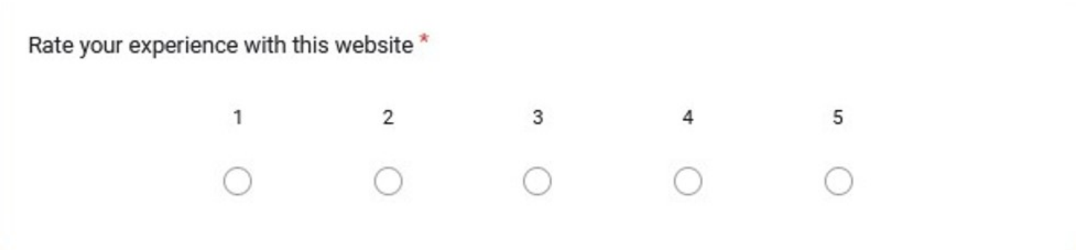


Figure A.16: Start Form Question 8

Appendix B

Post-Game Questionnaire

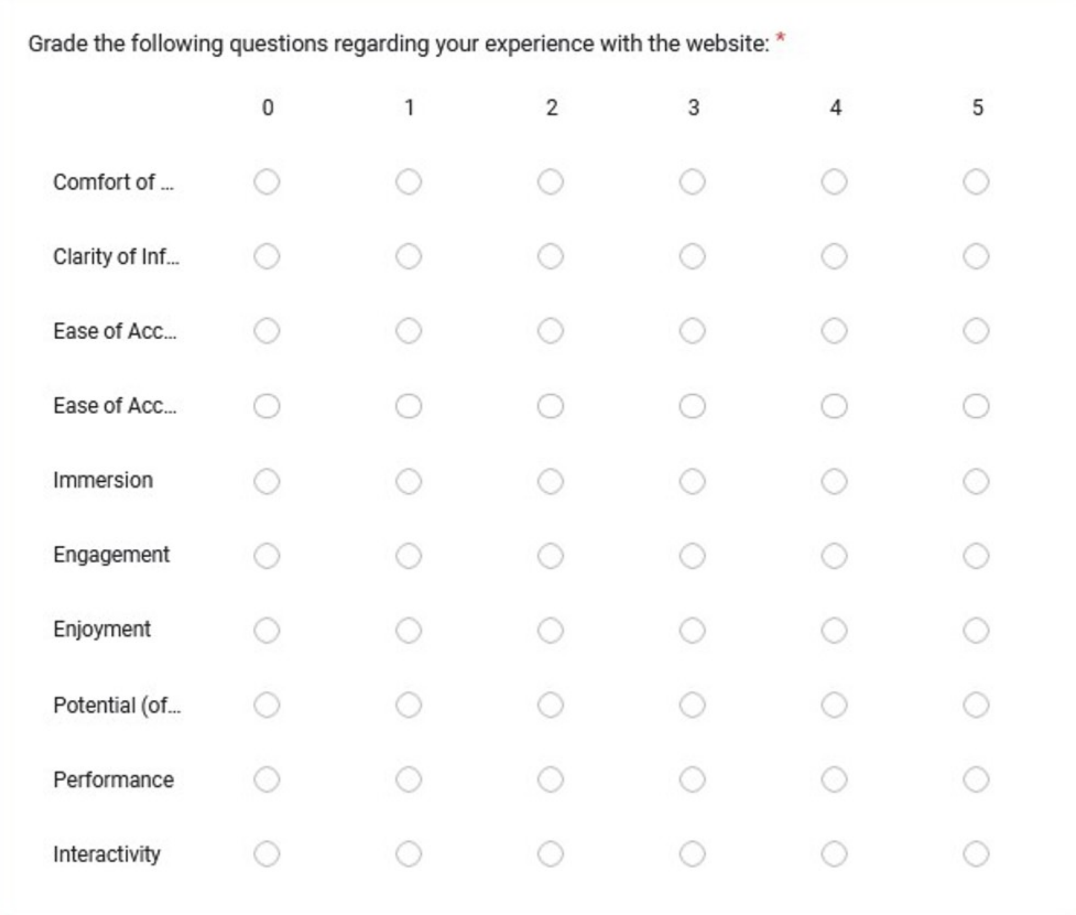
B.0.1 Questions



Rate your experience with this website *

1	2	3	4	5
○	○	○	○	○

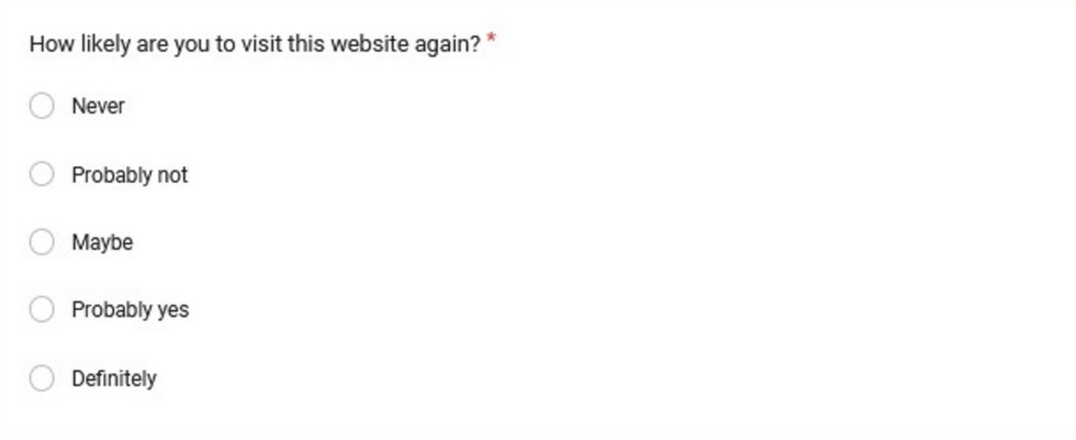
Figure B.1: End Form Question 1



Grade the following questions regarding your experience with the website: *

	0	1	2	3	4	5
Comfort of ...	○	○	○	○	○	○
Clarity of Inf...	○	○	○	○	○	○
Ease of Acc...	○	○	○	○	○	○
Ease of Acc...	○	○	○	○	○	○
Immersion	○	○	○	○	○	○
Engagement	○	○	○	○	○	○
Enjoyment	○	○	○	○	○	○
Potential (of...	○	○	○	○	○	○
Performance	○	○	○	○	○	○
Interactivity	○	○	○	○	○	○

Figure B.2: End Form Question 2



How likely are you to visit this website again? *

○ Never

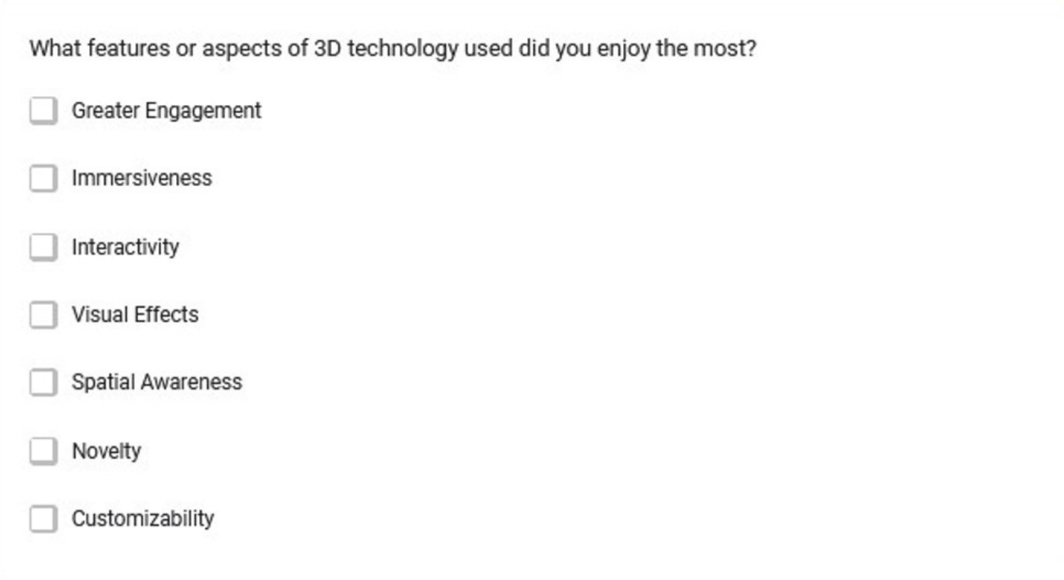
○ Probably not

○ Maybe

○ Probably yes

○ Definitely

Figure B.3: End Form Question 3



What features or aspects of 3D technology used did you enjoy the most?

□ Greater Engagement

□ Immersiveness

□ Interactivity

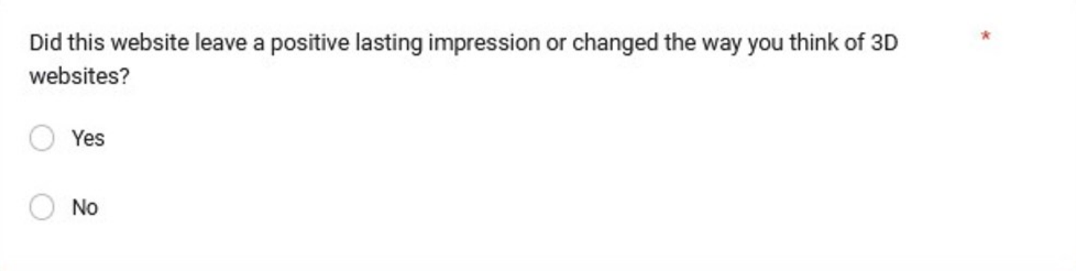
□ Visual Effects

□ Spatial Awareness

□ Novelty

□ Customizability

Figure B.4: End Form Question 4

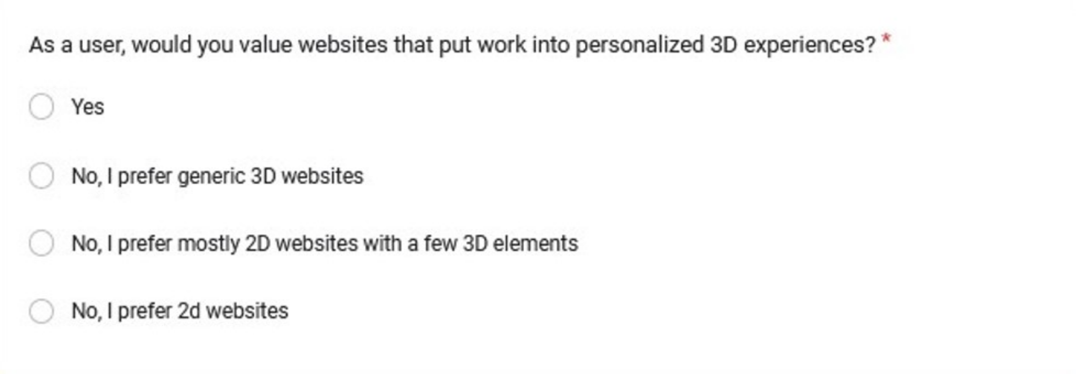


Did this website leave a positive lasting impression or changed the way you think of 3D websites? *

○ Yes

○ No

Figure B.5: End Form Question 5



As a user, would you value websites that put work into personalized 3D experiences? *

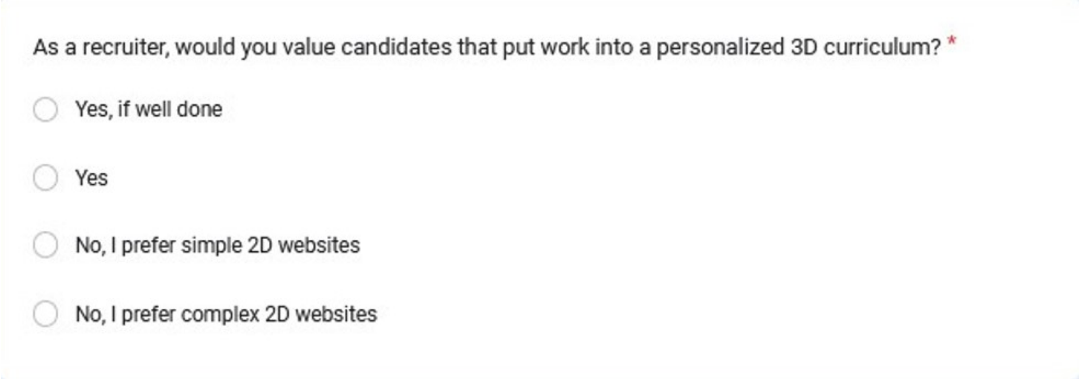
○ Yes

○ No, I prefer generic 3D websites

○ No, I prefer mostly 2D websites with a few 3D elements

○ No, I prefer 2d websites

Figure B.6: End Form Question 6



As a recruiter, would you value candidates that put work into a personalized 3D curriculum? *

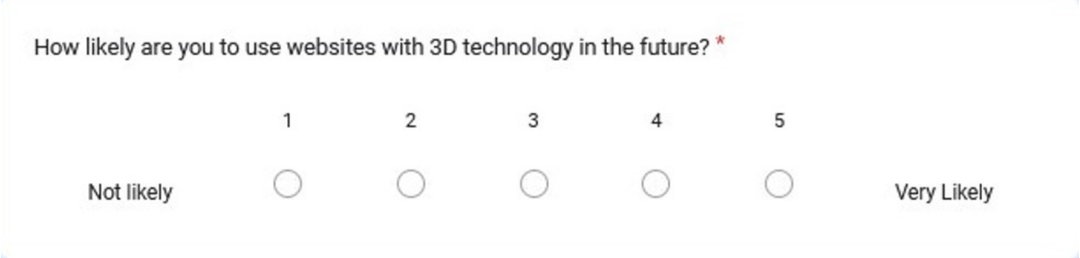
○ Yes, if well done

○ Yes

○ No, I prefer simple 2D websites

○ No, I prefer complex 2D websites

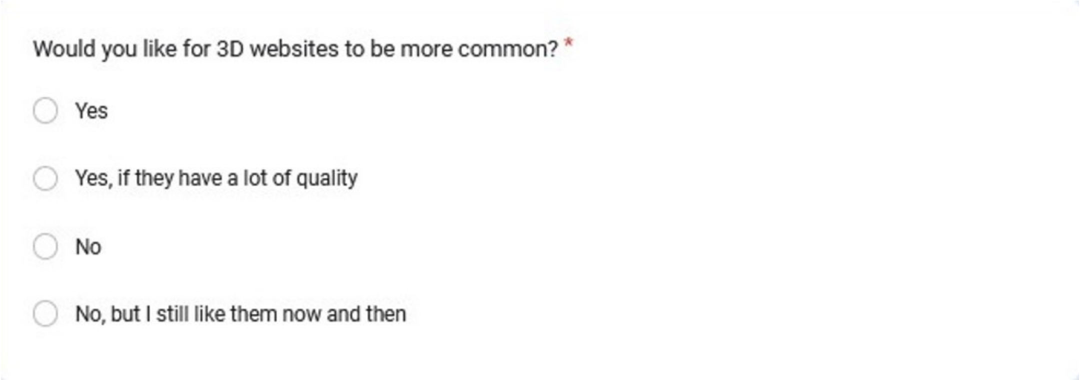
Figure B.7: End Form Question 7



How likely are you to use websites with 3D technology in the future? *

	1	2	3	4	5	
Not likely	○	○	○	○	○	Very Likely

Figure B.8: End Form Question 8



Would you like for 3D websites to be more common? *

○ Yes

○ Yes, if they have a lot of quality

○ No

○ No, but I still like them now and then

Figure B.9: End Form Question 9

B.0.2 Results

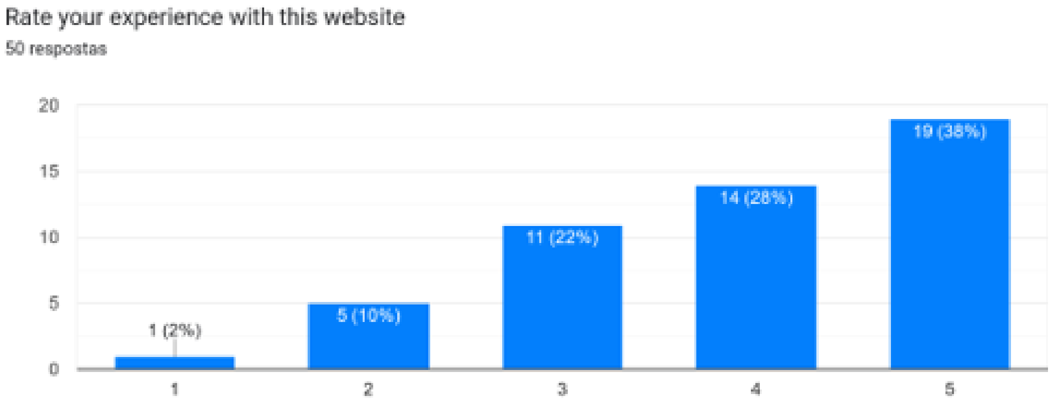


Figure B.10: End Form Result 1

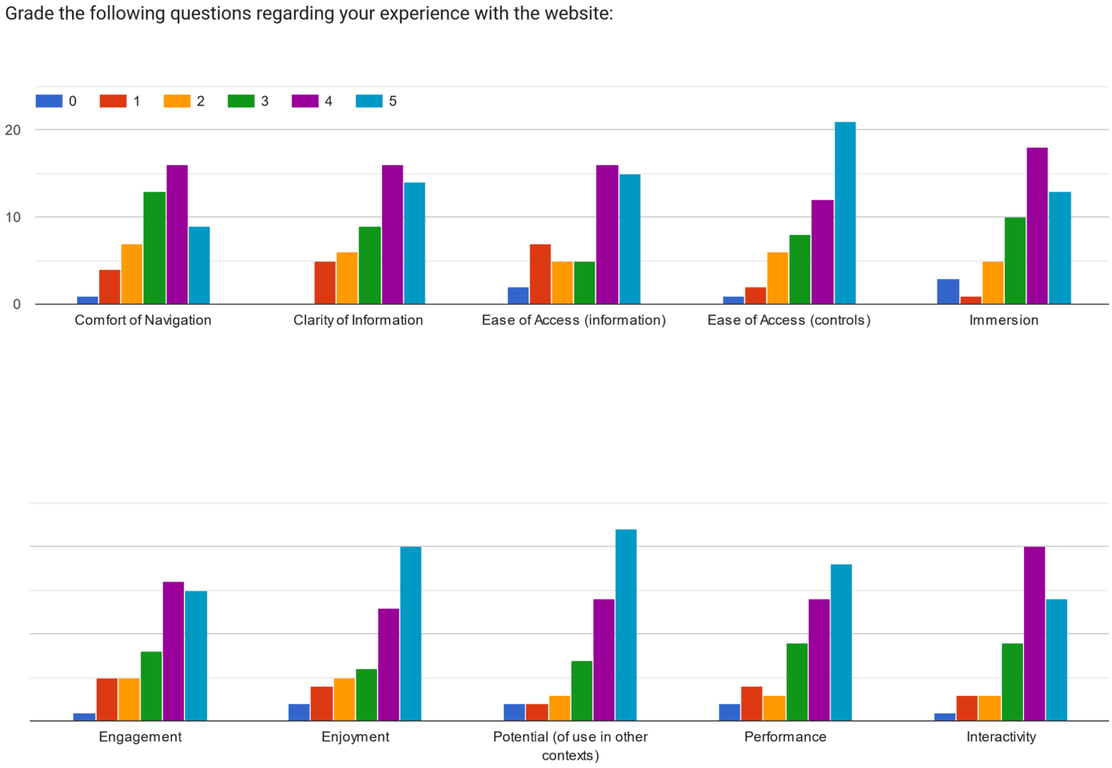


Figure B.11: End Form Result 2

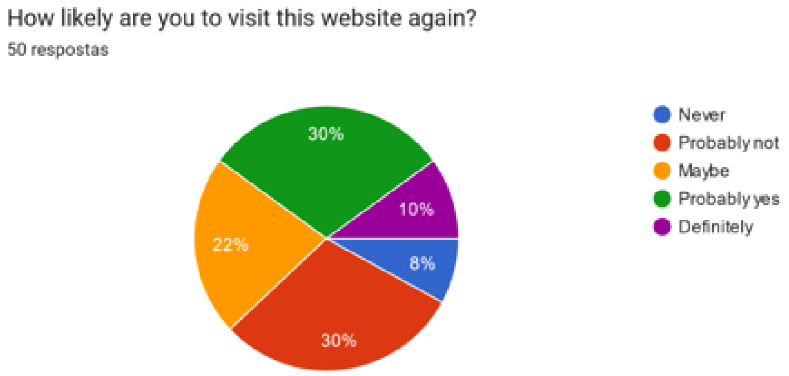


Figure B.12: End Form Result 3

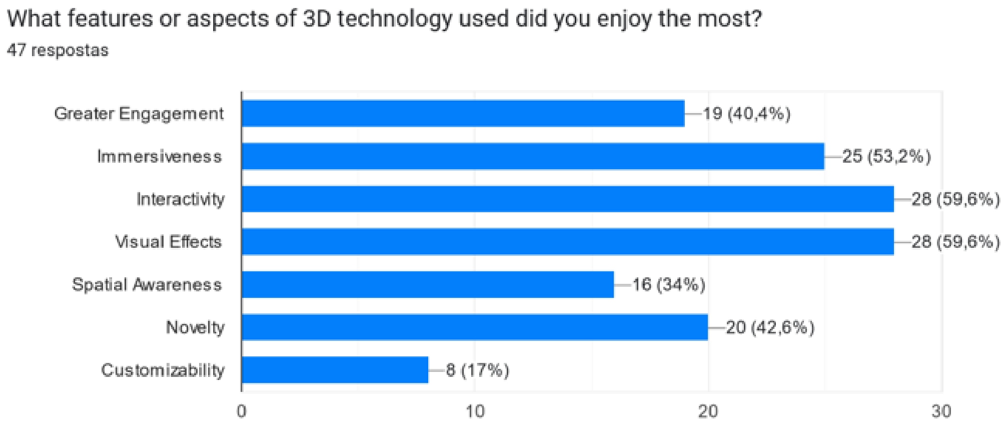


Figure B.13: End Form Result 4

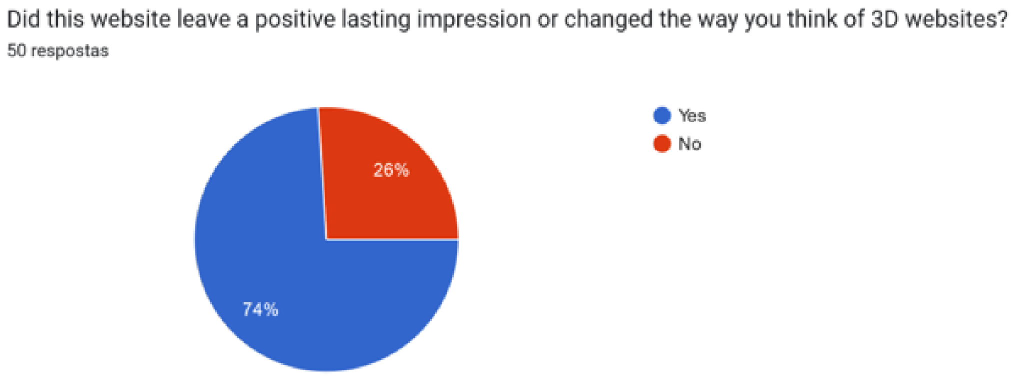


Figure B.14: End Form Result 5

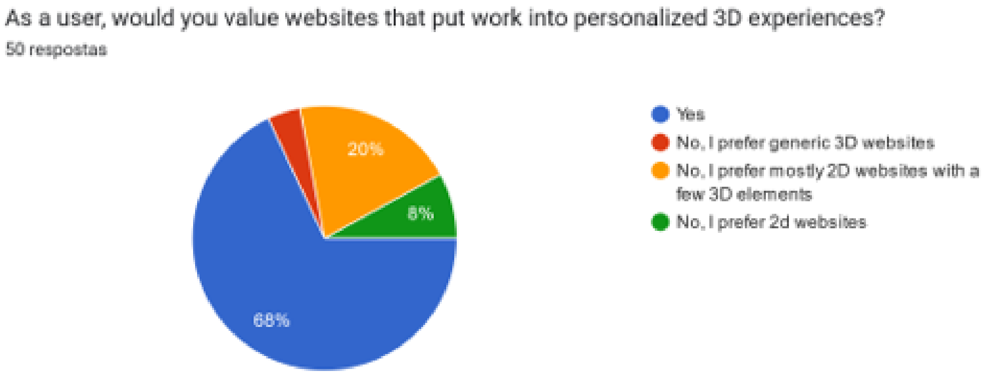


Figure B.15: End Form Result 6

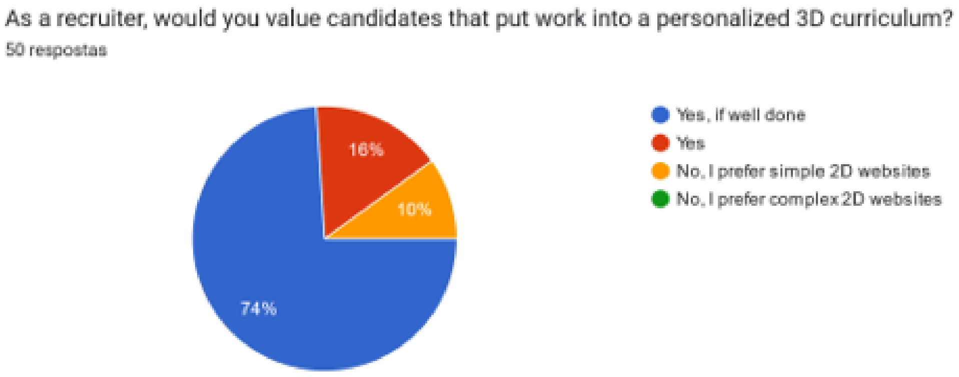


Figure B.16: End Form Result 7

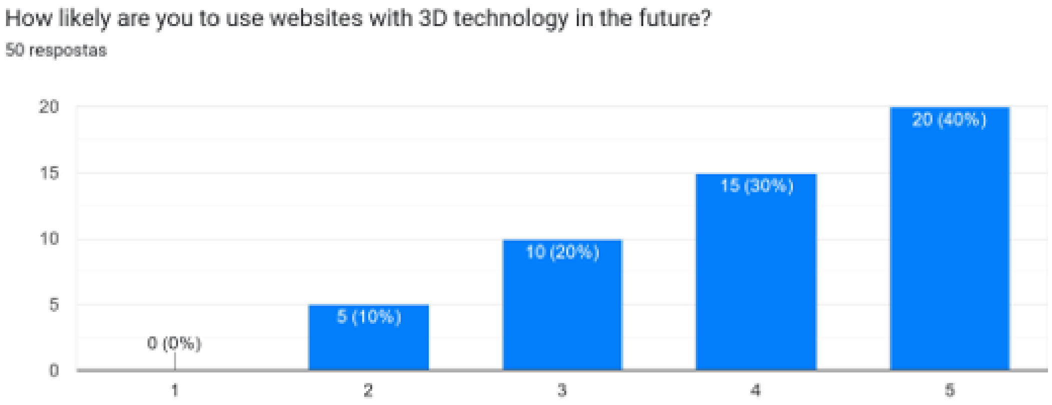


Figure B.17: End Form Result 8

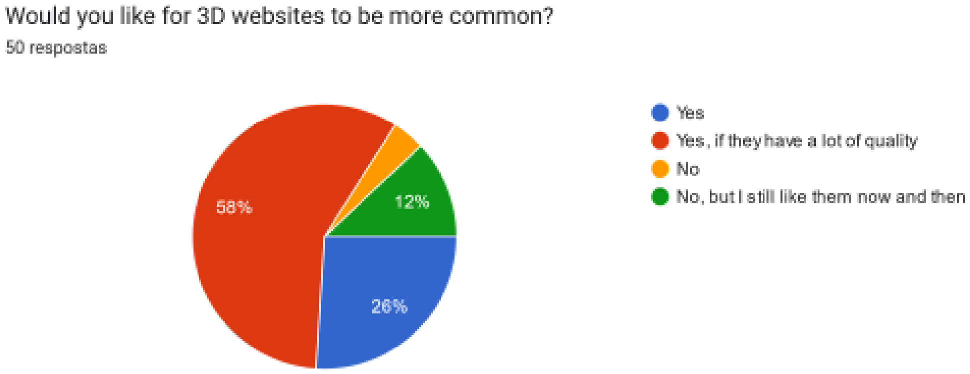


Figure B.18: End Form Result 9